Aalto University
School of Electrical Engineering
Degree Programme in Space Science and Technology

Leo Nyman

Suborbital Spaceplane Ascent Trajectory Optimization Using Simulated Annealing Technique

Master's Thesis
Espoo, April 10, 2017

Supervisor: Assistant Professor Jaan Praks
Advisor: Docent Antti Rasila

Aalto University
School of Electrical Engineering
Degree Programme in Space Science and Technology

ABSTRACT OF
MASTER'S THESIS

Author:	Leo Nyman		
Title:	Suborbital Spaceplane Ascent Trajectory Optimization Using Simulated Annealing Technique		
Date:	April 10, 2017	**Pages:**	79
Major:	Space Science and Technology	**Code:**	ELEC3039
Supervisor:	Assistant Professor Jaan Praks		
Advisor:	Docent Antti Rasila		

Suborbital spaceflight refers to a flight, which climbs above the Kármán line, but does not reach orbital velocity. The vehicle descents back to the dense parts of the atmosphere, before completing a full orbit. It is possible, that such flights will become more common in future, since market research has revealed demand for such a service. Increasing number of flights will result in increased demand for trajectory optimization and flight planning software solutions.

In this thesis, we will first have a look at the history of suborbital flight. Methods for simulation and modeling of such flights are then introduced. Next, we derive the cost function, which can be used to calculate a cost for a trajectory. This is done with multiobjective optimization, which allows one to set different weight factors to prioritize fuel or flight time savings.

The experimental part will first cover building of the simulation and modeling setup used in this thesis. Then, an algorithm based on simulated annealing technique is integrated into the setup. We test the operation of the algorithm using two test setups. The first test is designed to prioritize fuel savings. The second test aims to maximize the sum of kinetic and potential energies, at the end of an ascent trajectory. Based on the results, we conclude that simulated annealing is able to optimize a given trajectory, while taking into account the weighting factors for multiobjective optimization.

Keywords:	spaceflight, suborbital, trajectory optimization, simulated annealing
Language:	English

Aalto-yliopisto
Sähkötekniikan korkeakoulu
Avaruustieteen ja -tekniikan koulutusohjelma

DIPLOMITYÖN TIIVISTELMÄ

Tekijä:	Leo Nyman		
Työn nimi:	Aliorbitaalisen avaruuslentokoneen nousuvaiheen lentoradan optimointi simuloidun jäähdytyksen avulla		
Päiväys:	10. huhtikuuta 2017	**Sivumäärä:**	79
Pääaine:	Avaruustiede ja -tekniikka	**Koodi:**	ELEC3039
Valvoja:	Apulaisprofessori Jaan Praks		
Ohjaaja:	Dosentti Antti Rasila		

Aliorbitaalinen avaruuslento tarkoittaa lentoa, jolla noustaan Kármánin rajan yläpuolelle, mutta nopeus ei saavuta orbitaalista ratanopeutta. Alus laskeutuu takaisin ilmakehän tiheisiin osiin, ennen kokonaista ratakierrosta. On mahdollista, että tällaisten lentojen määrä tulee kasvamaan tulevaisuudessa, koska markkinatutkimukset osoittavat niille löytyvän kysyntää. Lentojen yleistyessä myös tarve lentoradan optimoinnille ja lennonsuunnitteluohjelmistoille kasvaa.

Tässä työssä luodaan ensin katsaus aliorbitaalisten lentojen historiaan, jonka jälkeen tarkastellaan kyseisten lentojen simulointiin ja mallinnukseen käytettäviä menetelmiä. Tämän jälkeen kuvataan kustannusfunktio, jolla mielivaltaiselle nousuvaiheen lentoradalle voidaan laskea kustannus monitavoiteoptimointia varten. Painotuskertoimien avulla optimoinnissa tullaan painottamaan erilaisia asioita, esimerkiksi polttoaineen tai lentoajan minimointia.

Kokeellisessa osassa kuvataan ensin tässä työssä käytetyn simulaation ja mallinnuksen rakentaminen. Siihen yhdistetään optimointia varten simuloidun jäähdytyksen algoritmi. Algoritmin toimintaa testataan kahdella koejärjestelyllä, joista ensimmäinen painottaa polttoaineen säästämistä ja toinen aluksen kineettisen ja potentiaalienergian summan maksimointia nousuvaiheen lopussa. Tuloksista havaitaan, että simuloitu jäähdytys kykenee optimoimaan lentorataa annetun painotuksen mukaisesti.

Asiasanat:	avaruuslento, aliorbitaalinen, lentoradan optimointi, simuloitu jäähdytys
Kieli:	Englanti

Acknowledgements

I thank my family for their wonderful support during my studies. I also express my gratitude to the Space Science and Technology faculty and students for maintaining space enthusiastic atmosphere here in Otaniemi. My deepest respect goes to the cosmonauts and astronauts who led the way to the Earth orbit and to the Moon in the 1960s. They have provided a wealth of inspiration to many people.

I am grateful to Jaan Praks for supervising this work, and to Antti Rasila for thesis related instructing.

I also thank everyone who helped with the proofreading and other support for this thesis.

Espoo, April 10, 2017

Leo Nyman

Nomenclature

Latin symbols

a	Speed of sound (m/s)
c	Thrust specific fuel consumption (g / kN $\cdot$ s)
D	Aerodynamic drag force (N)
$\mathbf{f}_{\mathrm{a}}$	Aerodynamic force (N)
$\mathbf{f}_{\mathrm{p}}$	Propulsive force (N)
$\mathbf{f}_{\mathrm{sp}}$	Specific force (acceleration, m/s^2)
g_0	Acceleration due to gravity at mean sea level. The value includes gravity and the opposite centrifugal component due to Earth's rotation. (m/s^2)
g	Local acceleration of gravity (m/s^2)
H	Altitude (m)
I_{sp}	See *specific impulse*
J	Performance measure for optimization. (dimensionless)
L	Aerodynamic lift force (N)
$\lim \dot{V}_{\mathrm{airframe}}$	Limit for acceleration of vehicle structure. (g-force)
$\lim \dot{V}_{\mathrm{crew}}$	Limit for acceleration based on human factors. (g-force)
m	Mass (kg)
m_i	Initial mass after takeoff at 500 m altitude. (kg)
m_f	Mass at the end of trajectory trace. (kg)
Ma	See Mach. (dimensionless)
$\mathbf{p}$	Control vector for engine staging
r_0	Radius of the Earth
S	Wing reference area. Sometimes defined so that it is overlapping part of the fuselage. (m^2)
T	Thrust (N) or the temperature variable for simulated annealing. [T] refers to a transformation matrix.
t_0	Time at the beginning of a simulation run. (s)
t_f	Time at the end of a simulation run. (s)

$\mathbf{U}$	Matrix $\mathbf{U}$ is used in this thesis to create control vectors $\mathbf{u}$ and $\mathbf{p}$ during optimization.
$\mathbf{u}$	Control vector for target true airspeed
$\mathbf{v}$	Velocity vector (m/s)
V	Velocity (referring to inertial velocity or velocity with reference to airmass, depending on the topic under study). (m/s)
W	Weight of the vehicle, $W = mg$. (N)

Greek symbols

α	Vehicle angle of attack relative to the free air stream (deg)
γ	Flight path angle (deg)
ρ	Air density (kg/m^3)
σ	State of the system which is being optimized

Coordinate systems and frames of reference

$]^{\mathrm{L}}$	North-east-down (NED) local-level coordinate system
$]^{\mathrm{V}}$	Velocity vector based coordinate system
$]^{\mathrm{B}}$	Vehicle body coordinate system
WGS 84	World geodetic system 1984

Acronyms and abbreviations

AOA	Vehicle's angle of attack relative to the free air stream
ASA	Adaptive simulated annealing
C.M.	Center of mass (Vehicle)
CADAC	Computer Aided Design of Aerospace Concepts (software)
CFD	Computational fluid dynamics
DATCOM	USAF stability and control toolkit
DOF	Degrees-of-freedom (3-DOF stands for three degrees of freedom)
GHAME	NASA's generic hypersonic aerodynamic model example
GS	Ground speed (m/s)
HTHL	Horizontal takeoff, horizontal landing
IAS	Indicated airspeed (m/s)
ISA	International standard atmosphere
LEO	Low Earth orbit
LH2	Liquid hydrogen

PCHIP	Piecewise cubic hermite interpolating polynomial
PID	A type of controller, which includes proportional, integral, and derivative terms in its loop
POST	Langley program to optimize simulated trajectory
RN	Reynolds number (dimensionless)
SA	Simulated annealing
TAS	True air speed (m/s)
TBCC	Turbine-based combined cycle engine
TDTP	Two-dimensional trajectory program
TPS	Thermal protection system
TSFC	Thrust specific fuel consumption

Terms used

Angle of attack	Same as α and AOA.
Boost-glide	A vehicle, which conducts powered flight only during the climb phase. Coasting phase and descent are unpowered.
Cost function	A mathematical formula used to predict the cost associated with a given set of inputs. The cost can have an arbitrary meaning.
Critical Mach number	The lowest speed at which some parts of the airflow around the vehicle reaches the speed of sound.
Downhill move	A move which lowers the value of the cost function, when searching for global minima.
Downrange	The horizontal distance traveled by the vehicle under study.
Gibbs free energy	Thermodynamic potential energy.
Global maximum	Highest possible value given by the cost function over all input values.
Global minimum	Lowest possible value given by the cost function over all input values.
Hypersonic	Any speed equal or higher than Mach 5.
Kármán line	By definition, the bordering line between space and lower altitudes.
Lifting-body	A vehicle with a fuselage shaped to produce most of the lift instead of wings.
Mach	Also Mach number. Represents the ratio of flow velocity to the local speed of sound.
Move class	A rule, which defines how the system can change its state from current state to a neighbouring state.

Performance measure	Same as *cost function*.
Point mass	A point in space, co-located with the vehicle's center of mass.
Positive-definite	Containing only positive values.
Uphill move	A move which increases the value of the cost function, when searching for global minima.
Ramjet	Aircraft engine which utilizes the vehicle's forward motion to compress intake air without a compressor. The fuel-air mixture is slowed to subsonic speeds for the combustion process.
Re-entry	Descent from space to the lower atmosphere.
Reynolds number	The ratio of inertial forces to viscous forces within a fluid.
Scramjet	Aircraft engine which utilizes the vehicle's forward motion to compress intake air without a compressor. The combustion process takes place in supersonic flow inside the engine.
Shuttlecock	A wing and stabilizer combination, which can be tilted to provide the required large drag increase during re-entry to the thick layers of the atmosphere.
Simulated annealing	A nature inspired Monte Carlo technique for solving optimization problems.
Spaceliner	A vehicle concept which carrier passengers and flies some parts of its mission above the Kármán line.
Specific impulse	Measure of the efficiency of rocket and jet engines.
State space	A collection of all the possible states of the system.
Subsonic	Any speed lower than the local speed of sound.
Supersonic	Speed higher than the local speed of sound and less than Mach 5.
Transonic	Condition of flight, where the air flow around the vehicle is partly subsonic and partly supersonic.
True north	Direction towards the geographic north pole.
Turbofan	Aircraft engine developed from the turbojet. Most of the thrust is produced by a large fan, driven by the turbine.
Turbojet	Aircraft engine which produces thrust mainly utilizing the exhaust from the turbine.
Wave drag	A component of the total drag. Caused by shockwaves originating from supersonic flow of air.
Wing-body	A vehicle concept, where most of the lift is being produced by the wings.

Contents

Chapter 1

Introduction

The airplane has proven to be an excellent invention for moving people and cargo around the world safely and quickly. Although the advances in speed and altitude have been small in the past 50 years, there is no physical barrier which prevents new types of hypersonic airplanes to be developed. These will allow long distance flights around the globe in a fraction of time taken by existing passenger jets. Because the altitude profile for such flights can reach the *Kármán line*, the vehicle is then operating in space by definition (von Kármán and Edson, 1967). For this reason we can call such vehicles *suborbital spaceplanes* or *spaceliners*. The flight is suborbital, as long as we do not reach or exceed the orbital velocity.

At hypersonic speeds, the vehicle's surface is subjected to severe aerothermal heating. This is caused by temperature increase in shockwaves and skin friction around the vehicle. Because the heating can reach hundreds or even thousands of degrees Celsius, the airplane surface and the overall thermal design must be able to routinely handle such heat loads (Jenkins and Landis, 2003, p. 12). Advances in materials science is finally providing light and durable heat-resistant composite materials, which should enable economically viable vehicle concepts to emerge (Stevenson, 2016). This in turn will lead to serial production of such vehicles, since there already exists a clear market demand (Savino et al., 2015). The phase of flight, where aerothermal heating is most severe, depends on the vehicle concept. A study conducted by Rockwell shows, that the most severe heating may occur during the ascent phase (Dietz et al., 1981, p. 1-24). This is contrary to the US Space Shuttle, where the most severe heating occurs during re-entry. At hypersonic speeds the aerothermal heating ensures a considerable engineering challenge (Anderson, 2007, Chapter 14). This is evident from the experience gathered during the Mach 5 and 6 runs with the experimental X-15 aircraft (Heppen-

heimer, 2009, p. 84).

Another key challenge is propulsion, because only airbreathing engines can provide good fuel efficiency when using hydrocarbon fuels. There is a constant drive to develop engines with a higher *specific impulse* (I_{sp}), since increased specific impulse means, that the engine produces more thrust per unit mass of consumed fuel. While the *scramjet* propulsion technology is promising, it is still in its infancy. Meanwhile, we can rely on the true-and-tested conventional rocket engine for hypersonic flight. Because rocket engines require, that we also carry the oxidizer in addition to the fuel, we should limit their use as much as possible, because of the oxidizer weight penalty. In practice, we can utilize turbofan and ramjet engines for flight in the dense parts of the atmosphere, and light up the rocket engine only when the speed and/or the altitude demand it.

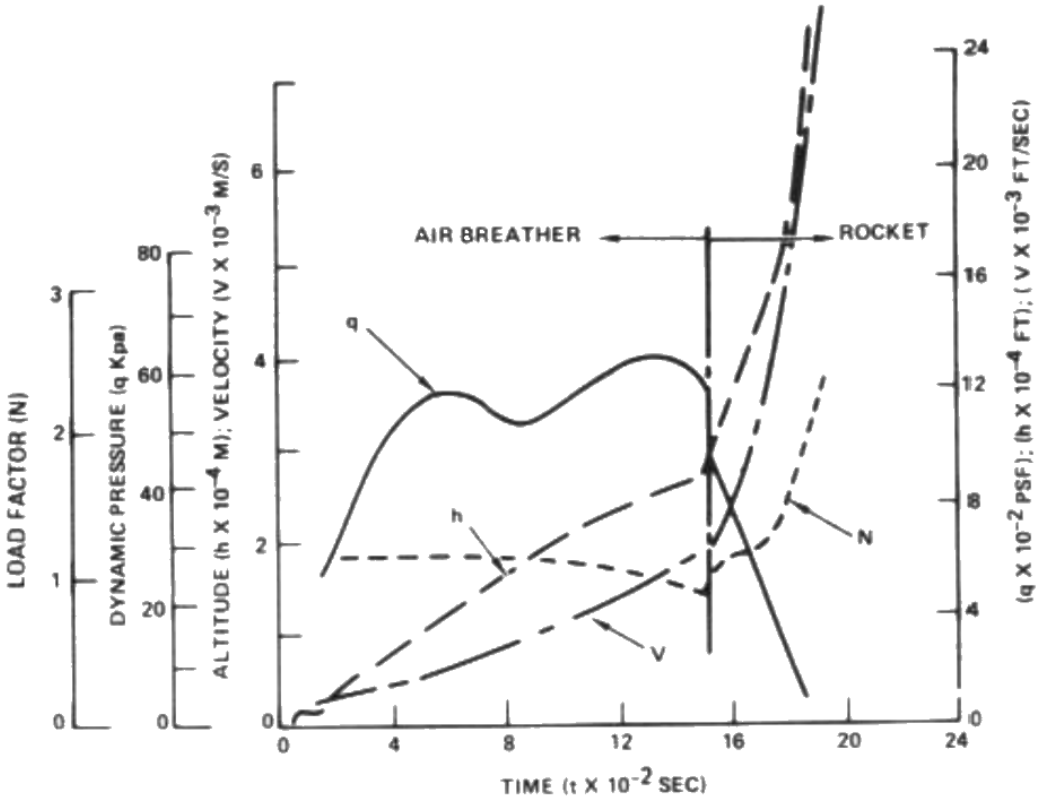

Figure 1.1: Ascent trajectory parameter histories for the Star-Raker vehicle concept (Reed et al., 1979).

Spaceplane ascent trajectories have been studied extensively since 1940s, when Sänger created his groundbreaking study (Sänger and Bredt, 1944). One good example is the *Star-Raker* horizontal takeoff, horizontal landing (HTHL) vehicle concept. HTHL vehicles were studied during the 1970s, as a technology which could have lowered the transportation costs to the low Earth orbit (LEO). This was seen as a key enabler, for the construction of large solar power stations into space. Figure 1.1 shows one analysis of the ascent phase for this type of vehicle. It was developed using two software tools, the *Two-Dimensional Trajectory Program* (TDTP) and the *Langley Program to Optimize Simulated Trajectory* (POST) (Reed et al., 1979). Recently, Betts has studied different numerical methods for aerospace trajectory optimization (Betts, 1998).

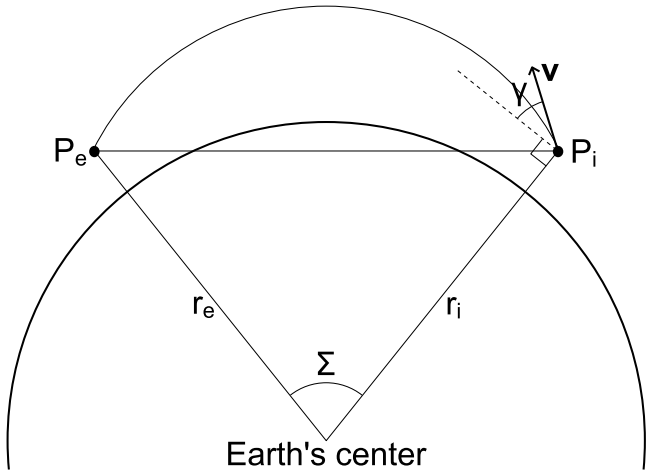

Figure 1.2: The ballistic arc for the coasting phase of flight, where P_i is the point of injection, P_e the point of entry, and r_i and r_e the distances to those points, measured from the center of the Earth. The vehicle's velocity vector $\mathbf{v}$ and flight path angle γ determine the resulting arc.

This thesis focuses on the powered phase of the ascent trajectory, in the relatively dense parts of the atmosphere. This phase of the flight ends in a *point of injection*, which marks the beginning of a coasting ballistic arc (Cornelisse et al., 1979, p. 295). The resulting angular range, for a given point of injection, is a function of the vehicle's velocity vector $\mathbf{v}$ and flight path angle γ (Figure 1.2). In this thesis, we have included the final velocity $\mathbf{v}_B^L$, as one of the state variables that contribute to the value of the *performance*

measure J. The notation used here is explained in Chapter 3. Using this concept, and the equations from available literature (Cornelisse et al., 1979), one can calculate the range of the suborbital arc, resulting from a given ascent trajectory. However, in this thesis we have not done so, but it is good to know the basics of ballistic arcs, in order to find desirable qualities for an ascent trajectory.

For suborbital spaceplanes, in some cases the ballistic arc is not purely elliptical, due to the effects caused by the atmosphere. If the gliding flight after the boost phase is conducted with the best lift-to-drag ratio in the influence of the atmosphere, the corresponding trajectory will be a wave-like skipping trajectory (Mack et al., 2011). To find the optimal glide parameters, refer to Vinh's book (Vinh, 1981, p. 328).

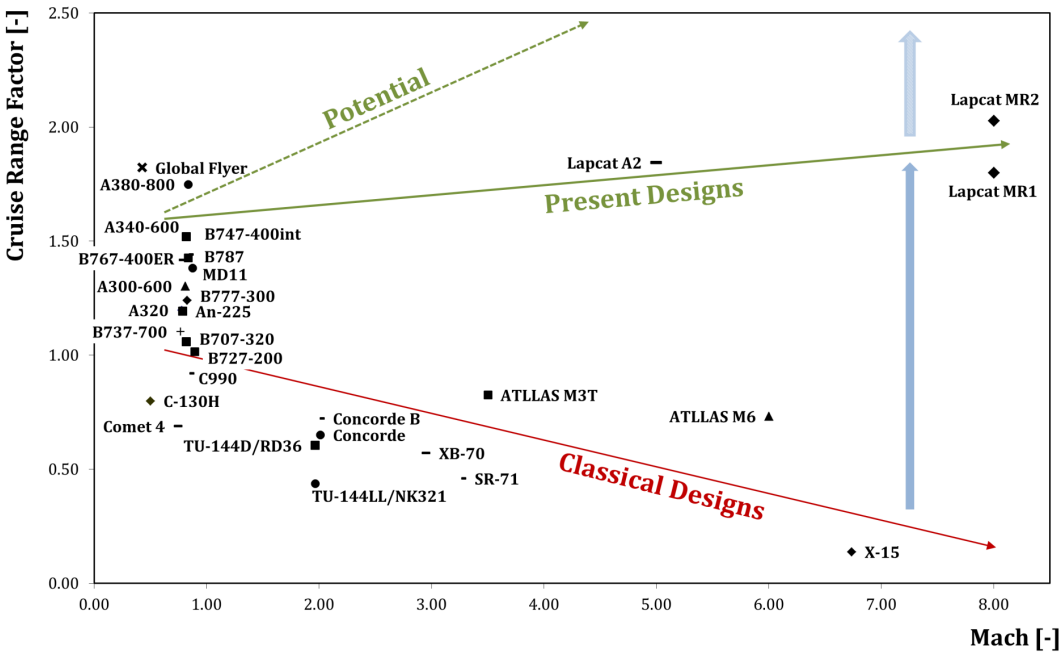

Figure 1.3: Long-range potential of new vehicle concepts (*Lapcat* and *ATLLAS*) as a function of Mach number. The plot also includes older designs, which have already flown. (Image: ESA)

The current interest in suborbital vehicles becomes clear, when one studies the key benefits. For example, a flight from Brussels to Sydney would take just three hours, with the fuel consumed per passenger per 100 km being the same as with current subsonic airliners (ESA, 2016*a*). The European Space Agency (ESA) has several ongoing technology development projects in this area. Figure 1.3 shows the *Lapcat* and *ATLLAS* concepts, plotted together with older designs, which have already flown.

Recent market survey shows, that there is a sizable market for suborbital flights (Le Goff and Moreau, 2013). Once suborbital spaceplanes start regular service, there should be an increase in the demand for flight planning and optimization services.

This thesis is meant to answer a specific question - Can we optimize the ascent trajectory using a *simulated annealing* (SA) technique? The motivation for this is clear. For a suborbital spaceplane, majority of the fuel is burned during the ascent phase of the flight. On the other hand, simulated annealing is a promising Monte Carlo technique for optimization tasks. It is simple to implement and it can find the *global minimum*, or a solution which is very close to the global minimum, in a predefined time. It is therefore easy to include in a computer software. Therefore, we are interested in finding out, whether we can combine trajectory simulation and simulated annealing, and whether the results are useful. There are few challenges with this. Simulated annealing is simple, but trajectory simulation is complex. Also, simulated annealing is a *technique*, which can be implemented in various ways in practice. It is difficult to find a particular type of implementation, which will be the best for our ascent trajectory optimization.

This thesis is looking for a way to use simulated annealing, to successfully optimize an ascent trajectory of a suborbital spaceplane.

This thesis is structured as follows. First, we will have a look at what the history can tell us. Then, one chapter is dedicated to the simulation of aerospace vehicles, operating in the atmosphere. Chapter 4 introduces relevant theory regarding optimization. Chapter 5 describes how the simulation and optimization engine was built for this thesis. Chapters 6 and 7 lay out the results and some analysis of them. At the end, we draw conclusions.

Chapter 2

Historical Perspectives

This thesis focuses on suborbital vehicles, which possess a high lift-to-drag ratio. Such vehicles, operating at the outskirts of the relevant layers of Earth's atmosphere, are commonly referred to as *spaceplanes*. Other suborbital vehicle types do exist, one good example being the Mercury *capsule*, used by Alan Shepard on his short suborbital "hop" in 1961 (Figure 2.1). His capsule, as well as Gagarin's Vostok capsule, which flew first and reached the orbit, had non-existent lift-to-drag ratios, therefore performing simple ballistic trajectories. Surprisingly, the later Apollo capsule had a significant lift-to-drag ratio of around 0.3 (Hillje, 1967). This enabled aerodynamic trajectory control during the re-entry phase. However, for the remainder of the text, we deal only with *wing-body* vehicles having the highest lift-to-drag ratios possible. This is because the lift-to-drag ratio is a crucial parameter for the range performance, giving excellent operational flexibility (Shevell, 1983, p. 210).

2.1 History and Theory of Suborbital Flight

In this section, we browse through the history and theory of suborbital flight in chronological order.

2.1.1 The Early Days

One of the first persons promoting spaceflight with a winged vehicle was Friedrich Zander, born in Latvia in 1887. Most of his influence was felt in the Soviet Union, were he gave lectures on the subject. He was convinced, that an aeroplane configuration was superior over basic rocket construction. This was based on the fact, that the aeroplane configuration is operationally more flexible, allowing gliding flight when the engine is temporarily or perma-

Figure 2.1: Alan Shepard - one of the best known suborbital pilots (Image: SuperStock)

nently shutdown. He also mentioned fuel savings during the descent phase, because the aeroplane configuration can use the planet's atmosphere to slow down its speed for landing (Hendrickx and Vis, 2007, p. 1).

The legendary Sergey Korolev worked actively with winged missiles during the 1930s. In 1935, he proposed a manned winged rocketplane, which could reach altitudes of 20-30 km (Hendrickx and Vis, 2007, p. 4).

One of the first comprehensive studies regarding suborbital flight with winged vehicles, was conducted by Sänger in Germany (Sänger and Bredt, 1944). His analyses, which included an example rocket powered aircraft, included climb profiles exceeding 100 km in altitude. The analyses pointed out, that the most efficient climb profile maximizes final velocity for a given fuel mass. However, the resulting trajectory requires acceleration values, which are unsafe for human occupants. This limits the set of acceptable trajectories.

Sänger initially makes the assumption, that five forces affect the climb, all acting through the center of mass, and in the same plane as the flight path:

- Weight of the aircraft. Direction towards the center of the Earth.
- Lift from aerodynamic lifting surfaces. Direction perpendicular to the tangent to the flight path.

- Drag due to the atmosphere. Direction is the tangent to the flight path.
- Thrust of the motor. Direction tilted from the tangent to the path by the *angle of attack*.
- D'Alembertian inertial force. Equal and opposite to the resultant of the other forces.

Because the Earth is rotating, and its atmosphere rotates with it, Sänger includes two more forces into the analysis:

- Coriolis force, due to the rotation of the Earth sphere and the associated reference frame.
- Pilot steering force. Equal and opposite to the apparent wind force, caused by the rotation of the atmosphere with the Earth. This steering force is required, in order for the flight path to follow the intended path, calculated by celestial navigation.

Sänger suggests, that one should first calculate the path using only the first five forces mentioned. Coriolis effect can be calculated separately, and its result can be used to correct the flight path. The most important datum, resulting from the climb analysis, is the resulting final velocity. It will affect the suborbital arc of the coasting flight phase (the engine remains shutoff during this phase).

After detailed study of Sänger's work, in 1947, Mstislav Keldysh created an improved concept, based on scramjet and rocket propulsion. This suborbital spaceplane would have flown similar "dip-and-skip" trajectories, as proposed by Sänger (Hendrickx and Vis, 2007, p. 18).

The next significant study was made by Dornberger (Bell Aircraft Corporation) in 1952 for the BOMI boost-glide vehicle (Jenkins and Landis, 2003, p. 12). This study added important new information regarding thermal problems for development of such vehicles. It also included potentially viable concepts to solve those problems. Later during the 1960s and 1970s Bell refined the concept, and introduced new propulsion and thermal control concepts (Helenbrook et al., 1971, pp. 3-4).

In the 1950s, US Air Force proposed a concept, where a Republic XF-103 would have been fitted with a ramjet engine to enable high-altitude Mach 5 flights (Trimble, 2016).

2.1.2 The Era of Space Exploration

The launch of the *Sputnik* satellite in 1957 signaled the beginning of a new era, the *space race*. Its focus was clearly in orbital flight, and in travel to other celestial bodies. In this section, we will instead have a look at what happened in the suborbital arena during that time.

Theodore von Kármán pointed out, that there exists a boundary, where the air density becomes so low, that the velocity required for sufficient aerodynamic lift is equal to the orbital velocity. This line is called the *Kármán line*, and by definition, it has been agreed to reside at exactly 100 km altitude. This concept was introduced in Kármán's autobiography, published in 1967 (von Kármán and Edson, 1967). The exact altitude can be calculated as follows. We first derive the aerodynamic lift force

$$L = \frac{1}{2}\rho S V_0^2 C_L, \tag{2.1}$$

where L denotes the lift force, ρ is the density of air at the given conditions, S is the reference wing area of the vehicle and C_L is the dimensionless coefficient of lift factor. V_0 denotes the speed of a circular orbit at the same altitude in vacuum. Then, using a quasi-static aircraft performance analysis, we find the altitude where the lift force L equals the weight of the vehicle $W = mg$.

In US, Peter Zipfel has been studying and developing techniques for aerospace vehicle trajectory simulations using computers since 1960s. He is also familiar with the earlier work conducted by Sänger and others. He has been promoting the use of the free CADAC software, which enables 3-, 5- and full 6-DOF simulations, as well as Monte Carlo runs (Zipfel, 2000).

X-15 was a highly successful hypersonic research aircraft, developed for the US Air Force by North American Aviation in cooperation with NACA (later NASA). Initial concept work was completed in mid 1950s, and the first unpowered gliding flight took place on 8th June 1959. Last test flight was flown on 24th October 1968, marking the end of a decade long research work (Jenkins and Landis, 2003, p. 229-248). The operational X-15 was always powered by a rocket engine, but there were proposals to build ramjet or scramjet powered versions. However, this type of hardware never flew (Jenkins and Landis, 2003, p. 215). There was a dummy ramjet model, which was attached to the ventral stabilizer on some of the flights. These tests revealed severe heating problems, caused by shockwaves originating from the

dummy ramjet (Jenkins and Landis, 2003, p. 145). During the project, several booster rocket equipped "X-15B" concepts were drafted. One of the concepts would have even been capable of orbital flight (Jenkins and Landis, 2003, p. 209).

Northtrop built the HL-10 and M2-F2 lifting-body experimental vehicles, which were test flown over 50 times between 1966 and 1975 in US. Their purpose was to study and validate safe flying and landing characteristics of such vehicles. They did not have a thermal protection system or other key systems, which would have enabled high-energy missions. Similar vehicle was also built by Martin (X-24A). Later more advanced X-24B and X-24C versions were designed. X-24B was test flown 36 times (Cobleigh, 1998, p. 5).

X-20 was to be an operational spaceplane, on contrary to the X-15, which served solely as a test platform. In 1963, the funding was cancelled, and the X-20 funds were reallocated to the Manned Orbiting Laboratory (MOL) project (Hendrickx and Vis, 2007, p. 18).

Although not a spaceplane, the North American XB-70 Valkyrie demonstrated a practical method of cooling the most heated surface regions in a Mach 3+ aircraft. Fuel from the fuel tanks was routed through heat exchangers placed at critical locations.

The Soviet Union produced its own share of spaceplane concepts. In the 1960s, Oleg Gurko started to promote his SSTO concept, which included a type of scramjet engine (Hendrickx and Vis, 2007, p. 447). Similar studies were conducted by Vladimir Myasishchev. Tupolev's design bureau created several boost-glide concepts mainly for military applications. Their work culminated in the 10-20 ton Zvezda vehicle, which would have flown its missions at altitudes of 50 to 100 km (Hendrickx and Vis, 2007, pp. 26-27).

Despite numerous concept studies performed in the Soviet Union, the only projects having actual hardware launched into space were Vladimir Chelomey's MP-1 and M-12 unmanned test vehicles (Hendrickx and Vis, 2007, p. 30). Most of the various spaceplane projects were cancelled during 1960s, when unmanned satellites and ICBMs proved effective in performing military tasks. The biggest project of its time, the 115-ton Spiral spaceplane designed by Mikoyan, never received official approval to proceed into production. After years of design work and some test flights with prototypes in the 1970s, the project was finally superseded by the Buran shuttle (Hendrickx and Vis, 2007, p. 30-37).

During the 1960s and 1970s, there were numerous small scale studies in US, focused on hypersonic flight with airbreathing propulsion. Some of these included scramjet engine concepts, instead of ramjets. There were hopes, that some of these vehicles would have evolved into SSTO capable operational systems, providing reliable and low-cost access to space. One such study was conducted by the US Air Force from 1959 to 1961 (Schweikart and Hallion, 1998, p. 13).

The US Space Shuttle and the Russian Buran shuttle were both capable of orbital flight. We can briefly mention them in this context, since they did have suborbital abort trajectories, to handle engine out scenarios and other emergencies. Both of these vehicles had their maiden flights during the 1980s.

Even after the US Space Shuttle procurement contract was signed, there were industry proposals for vehicles equipped with airbreathing engines. One interesting concept was the Rockwell *Star-Raker* reusable HTHL vehicle (Schweikart and Hallion, 1998, p. 18), which would have weighted a whopping 2300 tons during takeoff! According to the concept study, the vehicle would have been able to carry a payload of 100 tons into low Earth orbit, and was envisioned to enable the construction of large solar power stations into space. In addition to orbital missions, it was also envisioned to serve point-to-point transportation needs with suborbital flight profiles (Reed et al., 1979).

HOTOL was a British Aerospace concept, introduced to the public in 1984. The idea was to build a reusable spaceplane, capable of horizontal takeoff and landing (Conway, 2005, p. 209). One of the designers was Alan Bond, who is currently active in the Reaction Engines Skylon project.

President Reagan, after receiving advice from his science advisor George A. Keyworth II, introduced the Mach 25 hypersonic *Orient Express* concept to the public, in his 1986 State of the Union address. The concept was originally a secret DARPA project, conducted in cooperation with NASA and the US Air Force (Schweikart and Hallion, 1998, p. 1). The US Navy and the Strategic Defense Initiative Organization (SDIO) also participated, and the concept was named National Aero-Space Plane (NASP). Several concept studies were made, one of the most interesting being made by Du Pont in 1983 (Schweikart and Hallion, 1998, p. 23). The actual development project was conducted by NASA and selected aircraft manufacturers. Rockwell International's X-30 concept was developed further, and other vendors were placed

under Rockwell's coordination. Development continued, until the program was cancelled in 1993 due to budget cuts (Conway, 2005, p. 192). The NASP was envisioned to enable both suborbital and orbital missions. Some examples of suborbital missions included its use as a Federal Express cargo plane and a Mach 4 cruiser (Schweikart and Hallion, 1998, p. 56). A historic video can be found in Youtube (NASA, 2012).

The Russian counterpart to the Orient Express was the Tupolev Tu-2000. Development was cancelled due to budget cuts, after some sub-scale models had been tested by attaching them to rockets. In 2009, the Russian Sputnik news channel claimed, that the development has been restarted by government contract (Sputnik, 2009). The Tu-2000 is designed to use slush LH2 propellant, which has nearly 20% greater density compared to liquid LH2 (Haselbusch and McNelis, 1996). The greater density brings major benefits, since smaller fuel tanks are needed in the vehicle design.

Looking for a more efficient vehicle to replace the Shuttle, NASA requested proposals from major US aerospace companies in 1994. This resulted in interesting proposals from vendors. One of them was the McDonnell Douglas DC-X concept, which was actually test flown several times using a scaled model. Lack of government interest eventually mothballed this project, despite some of the results from the test flight campaign were very promising. End of the cold war resulted in less interest and funding for such vehicles. From among the industry proposals, NASA selected Lockheed Martin's lifting-body *VentureStar* design for further development. The contract included construction of a scaled down prototype named the X-33 (Reed and Lister, 1997). After test failures of the novel fuel tanks, US government cancelled the funding.

2.1.3 Suborbital Flight Development Projects in the 21st Century

As we proceed further into the 21st century, more and more hypersonic suborbital vehicle concepts are being constantly introduced. Also, big players such as Airbus, are developing their own concepts, but without full commitment to start serial production.

A big boost to suborbital flight occurred in 2004, when Mojave Aerospace Ventures flew their SpaceShipOne twice to an altitude exceeding 100 km. Having completed the second flight within the required time period of two weeks, the team secured the 10 million USD *Ansari X Prize.* SpaceShipOne

introduced a novel *shuttlecock* re-entry configuration, which is a working solution for this type of low-energy missions.

Suborbital mission profiles are classified into low-energy and high-energy transportation, depending on the downrange capability. Low-energy missions have a downrange of about 100 km, like in the SpaceShipOne and SpaceShipTwo. High-energy missions are more suitable for point-to-point transportation, covering thousands of kilometers of ground track (Mack et al., 2011).

Most suborbital vehicle concepts are based on a *wing-body* configuration. This enables a much greater down range distance, due to the higher lift-to-drag ratio, when compared against the *lifting-body* configuration.

Advances in materials science has introduced new viable options for aerospace engineers. In 2013, an Anglo-French research project demonstrated, that a carbonfibre-reinforced ceramic matrix composite material was able to withstand temperatures up to 1100°C (Stevenson, 2016). Such type of materials enable construction of lightweight thermal protection systems (TPS), required for hypersonic flight. High Mach numbers lead to high aerothermal heating, which is a problem especially for the nose, wind leading edges, vertical stabilizer and control surfaces of a winged hypersonic vehicle.

Some interesting ongoing projects are listed next. This is by no means a comprehensive list.

Founded by Sir Richard Branson, Virgin Galactic is a new aerospace company, aiming to start regular suborbital tourist flights to reach altitudes of approximately 100 km. The vehicle, *SpaceShipTwo*, has had a troubled development project. The first prototype was destroyed in a fatal accident in 2014. The second prototype is now conducting flight testing, and it has been partially redesigned to include enhanced safety features. The craft will be able to fly two pilots and six passengers to the edge of space (Norris, 2016).

Still in the drawing board, the UK based Bristol Spaceplanes Ascender is proposed to be a small two-seater suborbital spaceplane (Springs, 1998). It can reach similar altitudes as the SpaceShipOne, as it was also envisioned to compete for the Ansari X Prize.

The Airbus Defence and Space Spaceplane project was initially started at EADS Astrium, and is now coordinated under the Airbus brand. The vehi-

cle is envisioned to carry passengers or cargo on suborbital arcs, exceeding 100 km in altitude. A quarter scale model was test flown near Singapore in June 2014 (Ford, 2014).

HyPlane is an Italian aerospace concept for a six-passenger suborbital hypersonic business jet. It features airbreathing turbojet/ramjet engines, and a rocket engine for high altitude flight. It is planned to be used for both suborbital arcs for microgravity flights, and also for high altitude hypersonic point-to-point transportation. Trip distance of 6000 km would be covered in just 2 hours. Takeoff and landing would be possible using conventional runways and airports. An interesting feature of this plane is its use of a sharp wing leading edge, made possible by the use of modern high strength, high temperature materials such as boron carbide. HyPlane's performance capability is largely based on high efficiency aerodynamics and engine efficiency factors. For example, the CFD and DATCOM models show lift-to-drag ratios exceeding 4, even at a Mach number of 4.5 (Savino et al., 2015). The ramjet engines are envisioned to have a specific impulse (I_{sp}) from 1400 to 1600, which is a realistic goal by comparison to the engines used in the Lockheed SR-71.

Mach 6 capable SR-72 concept has been lobbied by Lockheed Martin in US. This vehicle would utilize a TBCC propulsion system (Trimble, 2016).

The European Commission has co-funded the FAST20XX project, to investigate critical technologies required for suborbital vehicles. It includes the *SpaceLiner* concept, originated by the German DLR.

Also, several other European technology development projects have been initiated to solve key challenges. These projects include for example the LAPCAT I & II, ATTLAS I & II, HEXAFLY, HEXAFLY-INT and similar projects conducted at national level. The HEXAFLY-INT project is studying a concept, where the re-entry into the dense part of the atmosphere is followed by a gliding flight at Mach numbers between 3 and 8 (ESA, 2016*a*).

The HIKARI project is a cooperation between European and Japanese space agencies (ESA & JAXA), and several industry partners. The goal is to develop a commercial spaceliner, capable of hypersonic speeds, and ready for regular service by 2050 (ESA, 2016*b*). Figure 2.2 shows one illustration of an aerodynamic analysis, made in the framework of the HIKARI project.

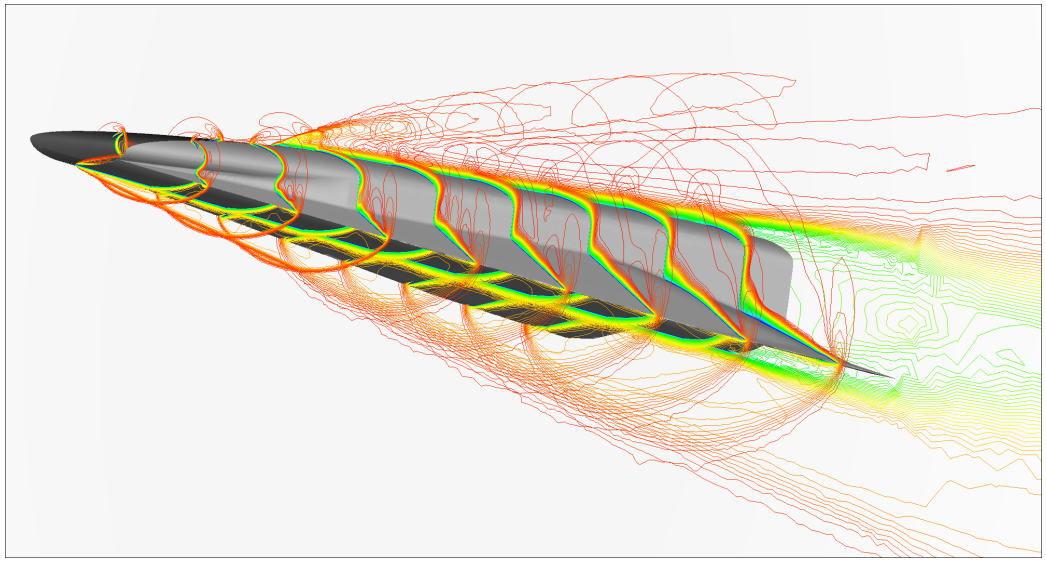

Figure 2.2: Computational illustration of a waverider concept at Mach 8 (Image: ESA)

2.2 History of Propulsion Concepts for Suborbital Vehicles

As we already noticed, there were suborbital vehicle concepts long before the turbojet or ramjet engines were invented. Therefore, the natural choice for many of the concepts, has been the rocket engine. It is still a viable option, as long as we need to rely on chemical energy systems.

One study conducted for the NASP project described a promising schema, where conventional propulsion was used up to Mach 2-3, ramjets at Mach 3-5 and a scramjet for the final ascent phase (Schweikart and Hallion, 1998, p. 31).

In 2004, the X-43 Hyper-X unmanned test vehicle achieved a historical 11 second scramjet powered flight. Aerospace engineers have extensively studied the theory of this type of propulsion, but operational scramjet engines have been very rare. During the X-43 test campaign, scramjet powered flight up to Mach 10 was demonstrated in November 2004 (Anderson, 2007, Chapter 14).

Recently, the Japanese Aerospace Exploration Agency (JAXA) has demonstrated a hypersonic pre-cooled turbojet engine, which looks promising for

high speed, high altitude flight (Taguchi et al., 2014).

For more details, Fortescue et al. (2005) contains a good overview of the aerospace propulsion systems developed to this date.

2.3 History of Trajectory Optimization

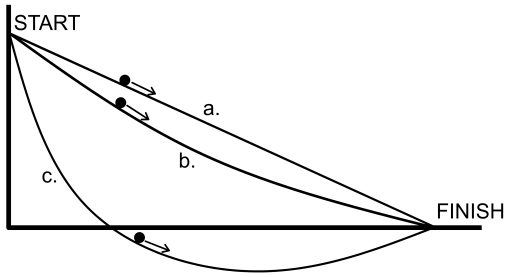

Figure 2.3: The *brachistochrone* problem, where the idea is to find an optimal path for a rolling steel ball. Curve c. is the optimal solution; hence the steel ball c. will reach the finish line first.

One very famous event in the history of trajectory optimization occurred, when Johann Bernoulli presented the *brachistochrone* problem to the public in 1696 (Figure 2.3). Isaac Newton received the problem by mail, after arriving home from work. By the next morning, he had already solved it! (Pars, 2010, p. 16).

Later, Jakob Bernoulli, who was Johann's brother, created a more demanding brachistochrone problem. The methods for solving it were later refined by Leonhard Euler, and became the *calculus of variations* (Pars, 2010, p. 17).

Early trajectory optimization work for aerospace applications was based on *indirect methods*, which requires that we first define necessary and sufficient conditions for optimality. These definitions require an analytical approach. One popular example of this method is the Pontryagin's maximum principle, based on the calculus of variations (Ewing, 1985, p. 321). One of the earliest studies was conducted by Hohmann (Hohmann, 1994). His trajectory optimization principle has been extensively used in space vehicle operations, where the atmosphere does not cause complications.

A special class of analysis is the so-called *energy state* method, for which several publications have been released since the 1950s (Rutowski, 2012).

Many analytical ascent trajectory optimization methods are aiming to minimize fuel expenditure, while maximizing final altitude and velocity. One good example is given by Krotov, Bukreev and Gurman (Krotov et al., 1971, p. 109).

Recent work on optimal control theory seems to promote *direct methods*, which do not require the complicated analytical work required with indirect methods. Betts mentions there are at least three major difficulties with the indirect methods, therefore his book focuses on the direct methods (Betts, 2001, p. 86).

Chapter 3

Simulation and Modeling of Atmospheric Flight

In this thesis, we are operating entirely within the Earth's atmosphere. Although approximately 99% of the total atmospheric mass is located below an altitude of 32 km, the thin portions extending up to 100 km altitude have considerable effect on our vehicle's performance, control and stability. Therefore, we must model the atmosphere at all times in our simulation. The atmosphere does not stop at 100 km altitude. It extends to much higher, but we leave that topic out of scope for this work.

It is very common to use a so called *standard atmosphere* model for aerospace concept studies and vehicle performance estimation. However, for operational applications we would rather have the most accurate real-world values for temperature, pressure and wind at different points along the planned trajectory. This is nowadays easy to arrange, with several supercomputer centers around the world providing weather datasets, describing the state of the atmosphere with unprecedented accuracy.

When we begin a task of building a flight simulation, it is important to first understand the concept of kinematics, before proceeding to study dynamics. When we combine masses, forces, Newton's & Euler's laws and kinematics, we can study the dynamics of aerospace vehicles (Zipfel, 2000, p. 87).

In this chapter, we first have a look at the basics of mathematical modeling, then we introduce important atmospheric concepts, and finally we go through relevant topics covering flight simulation.

3.1 Mathematical Modeling

A mathematical model is a simplification of a complicated phenomenon, constructed using mathematical terms and symbols. When a mathematical model is created, it can be used to study a real-world problem, typically with the help of computers (Figure 3.1). The process of using computers to solve the associated equations with initial values, parameters and variables, is called *simulation*. A simulation helps us understand, how the model behaves (Heiliö et al., 2016, p. 1). Mathematical modeling often requires

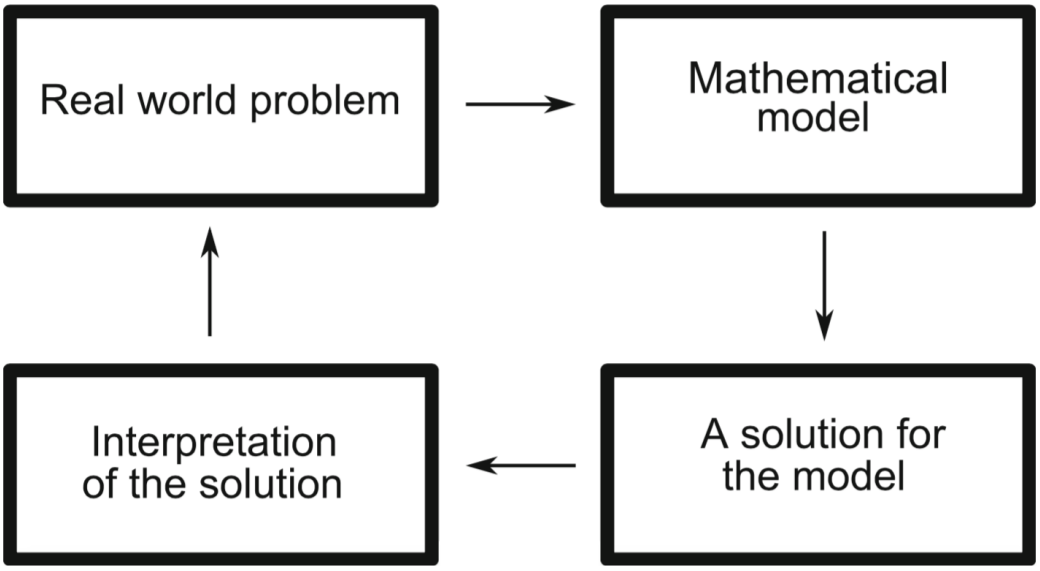

Figure 3.1: Typical phases of mathematical modelling (Source: Heiliö et al. (2016).

knowledge from several branches of science. Modeling and simulation enable a cost effective, and often a safer, method to study and develop complex real world systems. However, sometimes it is necessary to build real hardware and operate it, in order to collect datasets for improving and validating the model (Heiliö et al., 2016, p. 2).

Simplification of the model is often desirable, because it can enhance qualitative analysis/understanding of the system. It will also reduce the computational cost of the simulation (Heiliö et al., 2016, p. 171).

3.2 Modeling the Earth's Atmosphere

In order to input meaningful physical quantities into our simulation, we need to know what is needed. We could create a long list of items, for example by adding solar radiation and other phenomena. However, we should keep things as simple as possible, without losing too much accuracy. Therefore, only the most relevant ones are introduced here.

3.2.1 Atmospheric Motion

Atmospheric motion is a phenomenon well known to everyone as the *wind*. Vertical atmospheric motion can often be left out of simulations, without too much loss of accuracy. It can be substantial in the cores of thunderstorms and hurricanes, but aircraft are always routed well away from such dangerous areas.

Horizontal atmospheric motion can also have large velocities, but because the velocity gradient is typically shallow between large masses of air, it does not typically possess a danger to aircraft operations. In some cases, airlines fly inside jet streams on purpose, to benefit from the strong tailwind. When including weather datasets in the simulation, horizontal atmospheric motion will be included. Most of the active weather patterns in the world are contained in the troposphere. However, wind velocities can often exceed 45 m/s in the mesosphere (Müllemann and Lübken, 2005). This should be considered when selecting the source of the weather datasets.

The Earth rotates approximately 15 degrees per hour around its axis. This, combined with the radius of the Earth, results in a high tangential velocity at any given point on the surface, especially close to the equator. This raises the question whether the atmosphere follows this rotation, or is there a *slip*. The primary driver for winds in the atmosphere is not Earth's rotation. Instead, it is the uneven distribution of heat caused by absorption of solar irradiation. The rotation does have its effects, and the Coriolis force is one of them. Hence, for aerospace simulations, one can assume that the atmosphere rotates with the Earth. Wind velocity vectors, obtained from a database based on geographical location and altitude, simply describe the atmospheric motion with reference to an imaginary static atmosphere.

3.2.2 Physical Quantities for the Atmosphere

The standard atmosphere has a very distinct temperature profile. Starting from the sea level, as we move up, the temperature constantly decreases until we reach the tropopause. Then, in the lower parts of the stratosphere, temperature remains constant, only to start increasing in the higher parts of the stratosphere. Temperature starts to decrease again in the mesosphere, and then increases again in the thermosphere, due to absorption of solar radiation (Zipfel, 2000, p. 266).

The air pressure naturally decreases as we go to higher altitudes. Small regional variations in the pressure are caused by the low and high pressure weather systems, present mostly in the troposphere. It is common to provide the pressure and temperature data, which also enables one to calculate the density, using the *ideal gas law* (Zipfel, 2000, p. 266).

For high-fidelity simulations, the local acceleration of gravity g must be modeled accurately. The equation is readily available from literature (Shevell, 1983, p. 71). The value for g is calculated as

$$g = g_0 \Big(\frac{r_0}{r_0 + H} \Big)^2, \tag{3.1}$$

where g_0 is the local acceleration of gravity at mean sea level, r_0 is the radius of the Earth and H is the flight altitude.

3.2.3 Standard Atmosphere Models

There are several atmosphere models with varying altitude range and accuracy. The scientific community has also worked hard trying to agree, what is actually considered *a standard atmosphere*.

The simulation in this thesis is using the 1976 COESA United States standard lower atmospheric model included in Matlab. It covers the range from sea level up to 85 km geopotential altitude. For higher altitudes, the output values are extrapolated. The accuracy of the extrapolated values is sufficient for the purposes of this thesis. The Matlab subroutine, based on this atmospheric model, calculates pressure, density, temperature, Reynolds number and speed of sound as a function of altitude.

For simulations exceeding 85 km altitude, alternative or additional models can be incorporated. One such model is the NRLMSIS-00 model, which

takes into account data from the US Space Shuttle flights and new modeling techniques (Picone et al., 2001).

3.3 Modeling Atmospheric Flight

This section introduces some important concepts for modeling of atmospheric flight. For further reading, Vinh's book is a good starting point (Vinh, 1981).

3.3.1 Degrees of Freedom

Common examples of climb phase modeling use either three, five or six Degrees-of-Freedom (DOF) (Zipfel, 2000).

I have chosen 3-DOF, in order to keep the problem formulation simple, and to provide a computationally faster optimization engine. With 3-DOF, we treat the vehicle as a *point mass* (Heiliö et al., 2016, p. 130). For *performance* studies with aerospace vehicles, 3-DOF is sufficient without dilution of accuracy. If we were to study *control and stability* as well, 6-DOF simulation would be required.

3.3.2 Frames and Coordinate Systems

There are few mathematical methods, which are suitable for modeling how the vehicle climbs in the atmosphere. I chose to use tensors and transformation matrices, since they provide all the required tools. These tools enable us to model the path of the vehicle's center of mass. They also allow transformations between different frames of reference, such as Earth's inertial frame of reference and the body fixed frame of the vehicle.

Let's briefly discuss how frames and coordinate systems are linked. Each coordinate system has an associated frame, which has its 1, 2 and 3 axes aligned in parallel with the coordinate system's corresponding axes. Because frames have no origin, it is possible to define several different coordinate systems, that are aligned with the same frame. The opposite is not true, since different coordinate systems can have their origins at different locations. To keep things simple, we define just one coordinate system per frame in this work. Zipfel's book provides a good source of information regarding these topics (Zipfel, 2000).

Next, we have a look at the coordinate systems used in this thesis. Before that, some words about the notation. In some parts of this thesis, the equations are written using the notation promoted by Peter Zipfel (Zipfel, 2000, p. 22). This includes the axes labels, and the square brackets enclosing vectors and matrices. The basics of this notation follows. First of all, all 3D Cartesian coordinate systems follow the right-hand rule, and the axes are labeled as 1, 2 and 3. An axis label often includes a letter in superscript. This denotes the coordinate system, which is associated with the axis. Next section includes an example. For vectors, we can use an example from Equation 5.5, which contains the velocity vector $\mathbf{v}_B^L$. Zipfel's notation is always read from left to right or bottom to top. So, in this case, the velocity vector contains the velocity of the vehicle's body, with reference to the local-level coordinate system L.

The left to right convention can be explained when considering the transformation matrix $[T]^{LV}$, also part of Equation 5.5. In this case $[T]$ denotes a transformation matrix, and we read LV as "the local-level coordinate system with reference to the velocity coordinate system". Now that equation looks more intuitive. We notice that the transformation matrix handles a coordinate system transformation between the coordinate systems $]^{\mathrm{L}}$ and $]^{\mathrm{V}}$, and that here we also introduced an abbreviated notation to denote them. For more details, it's better to refer to Zipfel's book (Zipfel, 2000, p. 22).

3.3.2.1 Local-level Coordinate System

The main Cartesian coordinate system in which we operate, is the local-level coordinate system L, also known as the NED (North-East-Down) coordinate system (Figure 3.2). The 1^{L} axis is aligned with the true north, and the 2^{L} axis points east. Because we only want to use coordinate systems, which comply with the right-hand rule, the 3^{L} axis points down inconveniently. This means that some common variables such as altitude and vertical speed are negative, when the vehicle is climbing with its nose raised. We use this convention during integration, and replace the sign when plotting the results for easy interpretation. When using Zipfel's equations, flight path angle γ is positive for climbing flight, requiring careful approach, when used together with the local-level coordinate system (Zipfel, 2000, p. 81).

The origin is co-located with the projection of the point mass on Earth's surface. When the Earth is assumed a perfect sphere, the point mass, the projection point and the Earth's center are all aligned on the same line (the 3^{L} axis).

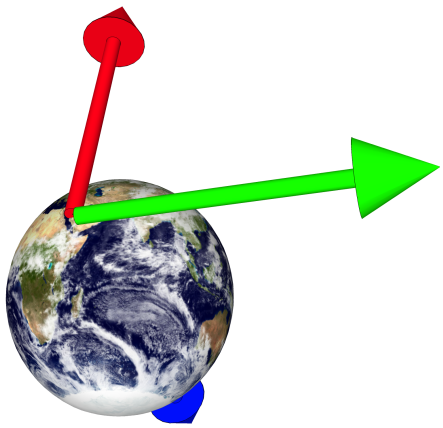

Figure 3.2: The local-level coordinate system. The axes are color coded as follows: 1^L (red), 2^L (green) and 3^L (blue). The 1^L axis is aligned with the true north, the 2^L axis points east. The remaining 3^L axis follows the right-hand rule and points downwards. This is a Cartesian coordinate system, having its origin co-located with the surface sphere of the Earth, and moving in connection to the vehicle position.

The origin and alignment of the axes can be linked to the WGS 84 coordinate system or equivalent, if one wishes to plot the position on a moving map display for example. The complications arising from the curvature of the Earth are avoided by using the *flat Earth approximation*. This is explained in Section 3.5.

3.3.2.2 Body Coordinate System

When the vehicle model incorporates AOA (angle of attack), we need to introduce the body coordinate system B (Figure 3.3). The 1^B axis is aligned with the vehicle's longitudinal axis and points to the nose of the vehicle. The 2^B axis projects along the right wing and the 3^B points down from the bottom of the vehicle, to conform with the right-hand rule. To be accurate, the 2^B axis is not governed by the wing construction, it simply follows the exact alignment of the 1^B and 3^B axes. In contrast, the 3^B axis is perfectly aligned with the vehicle symmetry, together with the 1^B axis. The origin is co-located with the point mass (Zipfel, 2000, p. 74).

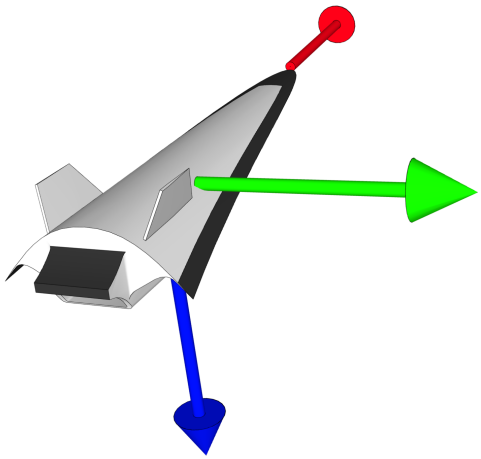

Figure 3.3: The body coordinate system, having its axes color coded as follows: 1^B (red), 2^B (green) and 3^B (blue). The 1^B axis is aligned with the vehicle's longitudinal axis, and lies together with the 3^B axis in the plane of symmetry. The 2^B axis extends in the general direction of the right wing, to conform with the right-hand rule. The origin is co-located with the C.M. *(Vehicle model by Riley Amos)*

3.3.2.3 Velocity Coordinate System

We need one more coordinate system, the velocity coordinate system, also known as the flight-path coordinate system (Zipfel, 2000, p. 79). It is easy to understand how the 1^B axis of the body frame is roughly parallel with the 1^V axis of the velocity frame, when the vehicle is in flight. Figure 3.4 shows an example, where the body frame's 1^B axis is tilted 15 degrees up relative to the 1^V axis of the velocity frame. If we assume a static atmosphere, that angle is equal to the AOA. It is important to realize, that the angle between the engine thrust vector and the 1^V axis changes whenever AOA changes. This effect must be included in the simulation, since the engine thrust is a significant force, and even minute changes in the angle will affect the resulting flight path.

3.3.3 Drag Polar

One key element in atmospheric flight is the *aerodynamic drag*. The concept of a *drag polar* was introduced as early as late 19th century, by the legendary Otto Lilienthal, who called it *die Flugpolare* (Shevell, 1983, p. 7). The fundamental idea is, that we can obtain the vehicle's drag coefficient C_D as a

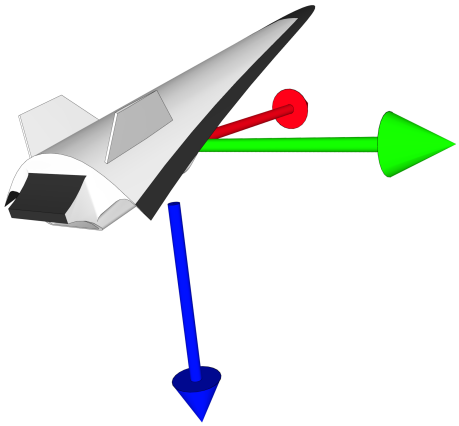

Figure 3.4: The velocity coordinate system, having its axes color coded as follows: 1^{V} (red), 2^{V} (green) and 3^{V} (blue). The 1^{V} axis is parallel and in the direction of vehicle C.M. motion. The 2^{V} axis is kept aligned with the horizontal plane and the 3^{V} points down with some amount of tilt, to conform with the right-hand rule. *(Vehicle model by Riley Amos)*

function of its lift coefficient C_L. The lift coefficient is not fixed, but varies based on several factors during flight. Therefore, the drag coefficient is also varying during the flight. We need to add one more independent variable, the *Mach number*, which enables us to accurately model the drag of a vehicle at supersonic or hypersonic speeds. To produce the highest level of accuracy, the Reynolds number should also be taken into account.

3.3.4 Aerodynamic Coefficients

Already in the early decades of aviation, it became a common practice to model aircraft aerodynamics using non-dimensional coefficients for lift (L) and drag (D). The coefficients are called C_L and C_D respectively. There are others as well, dealing with pitching moments for example, but those are not needed in our 3-DOF point mass treatment.

Lift force L from the aerodynamic reference area S, which consists of the wing area, and in some cases also parts of the fuselage, is calculated as

$$L = \frac{1}{2}\rho S V^2 C_L, \tag{3.2}$$

where C_L is the dimensionless coefficient of lift of the vehicle, ρ is the density of air and V is the velocity of the vehicle in relation to the surrounding air

mass (Shevell, 1983, p. 54). The force of drag is similarly defined as

$$D = \frac{1}{2}\rho S V^2 C_D, \tag{3.3}$$

where C_D is the dimensionless coefficient of drag of the vehicle. The coefficients of lift and drag can be derived from other known parameters as follows.

$$C_L = f(\alpha, Ma, RN) \tag{3.4}$$

$$C_D = f(\alpha, Ma, RN) \tag{3.5}$$

Here α is the vehicle AOA relative to the streaming airflow, Ma is the speed as a Mach number and RN is the dimensionless Reynolds number. As a simplification, the Reynolds number can often be left out, but shall be included in high fidelity simulations.

When α is increased, at some point the airflow will start to separate from the wing upper surface, eventually causing a stall. The highest C_L will occur just before this critical α angle and is denoted $C_{L_{max}}$. The corresponding α angle is known as $\alpha_{\max}$. These values can vary significantly at different Mach numbers. Therefore, $C_{L_{max}}$ must be defined as a function of Mach number (Vinh, 1981, p. 71).

For the normal safe operating speed range of an aircraft, and at given Mach and RN values, the relationship between α and C_L is linear (Vinh, 1981, p. 69).

It is very important to realize the dramatic increase in drag, which occurs in the transonic speed regime. The total value of the drag coefficient C_D is a combination of *wave drag*, *parasite* and *induced* drag effects, and its derivation is quite different for subsonic versus supersonic speed regimes (Shevell, 1983, p. 206). The parasite and wave drag are very much affected by the Mach number (Shevell, 1983, chapters 11 and 12). This is evident from the Figure 3.5, which shows the parasite drag as a function of Mach number. Anderson's book is a good source of information regarding drag at subsonic, supersonic and hypersonic speeds (Anderson, 2007).

3.3.5 Propulsion

When we mention engine thrust in this thesis, we always refer to the *net propulsive force*, which is the sum of the engine thrust, drag and other forces of the installed powerplant.

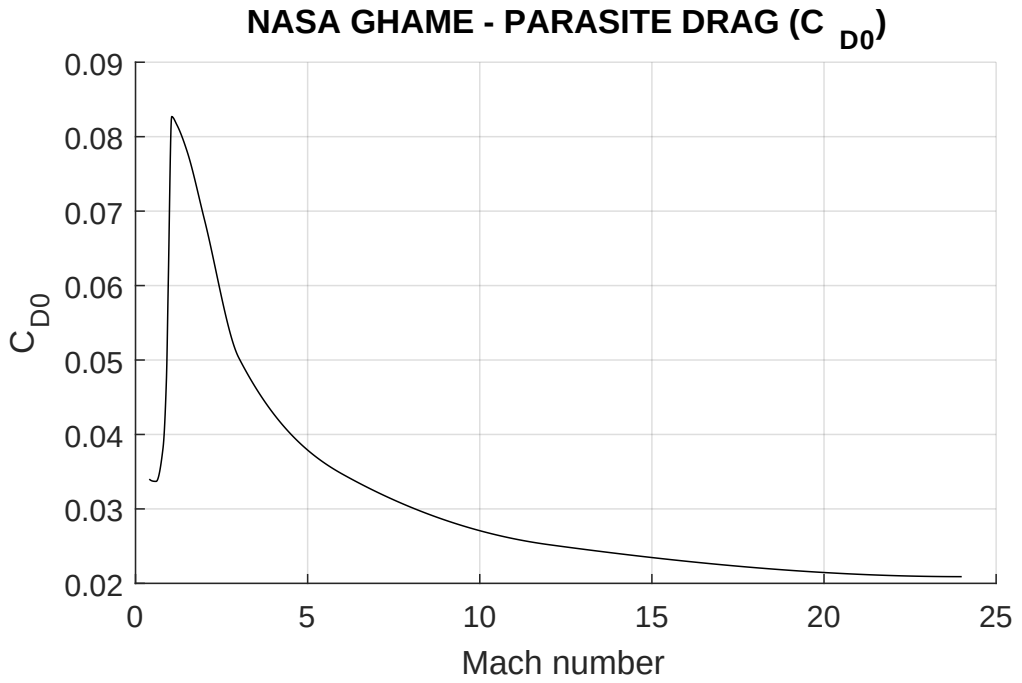

Figure 3.5: Parasite drag as a function of Mach number for NASA's GHAME concept vehicle. The X-axis shows the speed as a Mach number, and the Y-axis gives the corresponding aerodynamic drag coefficient. The Mach number, where abrupt drag rise occurs, is called the *drag divergence Mach number* (Shevell, 1983, p. 190).

The thrust equation for a turbofan engine combines equations for a turbojet and propeller engines, since the core of the engine operates as a pure jet, and the thrust given by the large fan can be modeled using propeller theory. Exact details can be found from literature (Shevell, 1983, p. 327).

For ramjets, Zipfel provides a useful method for determining the available thrust (Zipfel, 2000, p. 276).

The thrust of a rocket engine is relatively simple to calculate, since the engine manufacturer provides the relevant specific impulse I_{sp} values. Rocket thrust is calculated as

$$T = c\frac{\mathrm{d}m}{\mathrm{d}t}, \tag{3.6}$$

where T is the thrust, c is the thrust specific fuel consumption and the last term is the mass of consumed fuel per unit time (Shevell, 1983, p. 344).

In this thesis, NASA's EngineSim version 1.8a software was used to build the thrust and TSFC training datasets for the turbofan and ramjet engines. These datasets were then used to build 2nd order polynomial multivariate regression models, used in the Matlab code. The models used for this thesis did not include *cubic splines*. By adding the splines, the accuracy of the model can be increased, with the expense of added complexity (Heiliö et al., 2016, p. 60). The thrust for the rocket engine was calculated using the Equation 3.6.

3.4 Mach Number

Mach number is a convenient unit when representing vehicle speeds approaching the speed of sound (Mach 1), and for speeds above that. When a subsonic vehicle accelerates towards Mach 1, a phenomenon called the *wave drag* starts to act heavily, when a speed called the *critical Mach number* is reached. There is a noticeable drag peak around Mach 1, which reduces noticeably after around Mach 1.2, if the vehicle is further accelerated. For this reason, all air vehicles try to avoid operating in the transonic speed range for long periods of time. It also has a major impact on engine selection, since the thrust required to push through the "sound barrier" is significant.

The *true air speed* (TAS) can vary substantially for a constant Mach number, depending on the air temperature, which varies according to the altitude and local or seasonal weather patterns (Shevell, 1983, p. 95).

Supersonic Mach numbers are in the range of Mach 1 to 5. Most suborbital vehicles flying low-energy missions operate in this speed range. Suborbital vehicles flying high-energy missions, will most probably operate some parts of their trajectory at hypersonic speeds, which is by definition Mach 5 and faster. Very high Mach numbers cause severe aerodynamic loads at low altitudes, caused by the high dynamic pressure. For mesosphere and higher altitudes, air density is so low that aerodynamic loads are usually not an issue. The same is not true for aerothermal heating, which must the taken into account throughout the trajectory. The upper speed for any suborbital missions is around Mach 25, which is a sufficient final velocity for reaching the low Earth orbit (LEO) (Schweikart and Hallion, 1998, p. 62).

3.5 Flat Earth Approximation

It is often desirable to include simplifications in our work, but care must be taken to ensure sufficient accuracy in the results. There is a famous quote from Einstein, *"Everything should be made as simple as possible, but not simpler"*.

For space vehicles reaching orbital speeds, we shall use spherical coordinate systems. For suborbital vehicles, we are allowed to unwrap the Earth's surface into a plane, and use a more common 3D Cartesian coordinate system (Zipfel, 2000, p. 368). This method is known as the *flat earth approximation*. It is a common practice to use the flat earth approximation in studies, where the resulting error is negligible. The error resulting from this approximation can be calculated using the well-known *arc length from radians* formula. The required correction factor is calculated as

$$\frac{H + r_0}{r_0}, \tag{3.7}$$

where H is the geometric flight altitude in meters and r_0 is the mean radius of the Earth's surface.

For this thesis, the commonly used local-level coordinate system was selected as the base coordinate system for simulation. Therefore, we have made the choice of using the flat earth approximation to simplify our work.

Chapter 4

Optimal Control Theory

According to Betts, we can divide optimal control problems into two classes, *direct methods* and *indirect methods* (Betts, 2001, p. 85). In this section, we first look at the indirect methods, which require an analytical approach. In this case, one needs to analytically derive the necessary conditions for optimality. On the contrary, the direct method is often easier to implement, but requires sufficient computing power. In this thesis, we are using simulated annealing, which belongs to the class of direct methods.

4.1 Analytical Solutions

Analytical solutions are possible, if we work with the static ISA atmosphere model. A good example can be found from the work done by Ardema,Windhorst and Phillips (Ardema et al., 1998).

The *Breguet range equation* is one of the oldest aircraft performance equations (Shevell, 1983, p. 272). It can be written in the form

$$R = \frac{aM}{c}\frac{C_L}{C_D}\ln\frac{W_1}{W_2}, \tag{4.1}$$

where R contains the range, a is the local speed of sound, c is the *thrust specific fuel consumption* (TSFC), M is the Mach number and W_1 and W_2 the initial and final aircraft weight respectively. Note that the variable names used in this equation do not follow the convention otherwise used in this thesis. The Breguet range equation can be used for a flight with a constant angle of attack and Mach number. Therefore, it is useful for a constrained analysis, where these two values are invariable.

Due to the limitations of existing methods at that time, in 1953, Edward Rutowski introduced the *energy state* method for ascent optimization of high performance aircraft. This method covers the full range of possible altitudes and speeds (Mach number) during an ascent phase. The instantaneous energy state of the vehicle is the sum of its kinetic and potential energies. The total energy can then be used as a variable in the optimization, which can be set to minimize fuel expenditure or flight time. When accelerating from subsonic to supersonic speeds, the optimum climb profile often includes a counter-intuitive diving section, where the aircraft dives and accelerates through the transonic speed regime. Using the energy state method, it can be proven that this is beneficial, since the time spent in the transonic speed regime is minimized (Rutowski, 2012). Figure 3.5 illustrates and gives a clear motivation, why exposure to the transonic speed regime should be minimized.

4.2 Simulated Annealing

Simulated annealing (SA) is a technique, which is well suited for combinatorial optimization problems (Kirkpatrick et al., 1983). When applied successfully, it can find the optimum or near-optimum solution for a function from a finite set of independent variables, by using a pre-defined number of computing cycles. This makes it a good option for many software applications, where computing time has finite constraints. The SA is inspired by nature. It is based on statistical mechanics and condensed matter physics, which are used to model how atoms in liquids and solid matter behave, when temperature is cooled down using a pre-defined schedule. Such controlled cooling is used, for example, to create highly isotropic atomic lattices from silicon. The resulting silicon crystal then contains the minimum amount of potential energy, also known as the *Gibbs free energy*.

According to Schneider and Kirkpatrick, the SA technique emerged through the work of several scientist solving optimization problems. They mention at least the following authors, Kirkpatrick, Gelatt, Wilson, Cerny, Metropolis and Ulam. However, Gelatt and Kirkpatrick were the first to formulate the SA in a generic form, which could then be applied to a wide range of optimization problems (Schneider and Kirkpatrick, 2006, pp. 80-81).

4.2.1 Derivation of Simulated Annealing

SA simulates the cooling process of a physical system, where temperature control plays a crucial role. By following a certain cooling schema, it is possible to form flawless crystals, which have the minimum amount of Gibbs free energy. In other words, the physical system ends up in an optimum state. Rapid quenching (cooling) of the system, will only bring the system to some local minima, giving proof to the fact that the temperature control plays an important role. On some systems, a sharp transition at some temperature can be observed. In materials science, this is called the *Curie* temperature. Good examples are the melting and boiling temperatures, where a material completely changes its behavior.

At each temperature step, the state σ of the system is changed several times. Successive states are reached by following a *Markov process*, where each transition $\sigma_i \rightarrow \sigma_{i+1}$ has a certain transition probability $P(\sigma_i \rightarrow \sigma_{i+1})$. For classical physical systems, the distribution of these transitions is the Boltzmann distribution. Now, considering that τ is another possible state, we assume that there exists an equilibrium of transitions, satisfying

$$\frac{P(\sigma \rightarrow \tau)}{P(\tau \rightarrow \sigma)} = e^{\frac{-\Delta E}{k_B T}}, \tag{4.2}$$

where E is the energy of the complete system, k_B is the Boltzmann constant and T is the temperature, all in SI units. Because states with a high potential energy have smaller probability, the Boltzmann equilibrium of the system ensures (quasi) optimality, as the temperature gradually cools and the equilibrium state converges towards the global minimum at zero temperature (Schneider and Kirkpatrick, 2006, pp. 81-84).

4.2.2 Acceptance Probability

SA accepts a change of state in the system, if a move reduces the total "energy" of the complete system. This is also termed a *downhill move*. If the move does not change the total energy, but we still accept the move, we are then applying a so-called *Metropolis criterion* (Schneider and Kirkpatrick, 2006, p. 84). There is no generic rule when to use this criterion, it will depend on the application. SA treats *uphill moves* differently. Uphill moves are accepted based on probability

$$P(\Delta E) = x^{\Delta E}, \tag{4.3}$$

where $x \in [0, 1]$ is a parameter. Typically $P(\Delta E)$ is compared against a value taken from a pseudorandom generator, which gives a number uniform on $[0, 1]$. As an example, if $P(\Delta E) = 1$, then all uphill moves are accepted. Another method is to define $P(\Delta E)$ like it was defined in the *Metropolis algorithm* (Kirkpatrick et al., 1983). In this case, the probability is calculated as

$$P(\Delta E) = e^{\frac{-\Delta E}{k_B T}}. \tag{4.4}$$

4.2.3 Important Simulated Annealing Parameters

One of the most important parameters for the SA, is the *cooling schedule*. Nourani and Andresen have compared different schedules in their paper (Nourani and Andresen, 1998). Another parameter, which has a huge impact on the computing time, is the choice of a *move class*. The move class specifies the rule, which controls how the current *state* of the system can change to yield a new "mutation". The allowed set of these states is called the *neighbours of a state*. The *state space* contains all the possible states.

4.2.4 Naming Conventions

In optimization, we often refer to the value of an *objective function*, also known as a *cost function*. In SA we often refer to *energy*, because of how the SA technique was originally derived. In this thesis, we replace the term *energy* with *performance measure* J. A list of common SA related terms can be found from the work of Salamon, Sibani and Frost (Salamon et al., 2002, p. 17).

Chapter 5

Setting up the Simulation

This chapter explains, how the problem was formulated and how the required simulation was built, to enable the use of the simulated annealing (SA) technique for optimization tasks.

5.1 Problem Formulation

This thesis covers an ascent trajectory optimization. This is not yet a sufficient statement, for defining the problem that we are trying to solve. Different customers have different needs (Shevell, 1983, p. 260). For some, time is not so important, and they prefer to save as much fuel as possible. For others, it might be the opposite. In many cases, the desired flight profile is a compromise between fuel burn, time and other factors, such as passenger comfort and safety. In this work, we treat safety and comfort as constraints, and make sure these limits are not exceeded.

Next, we need to define, what is a good ascent trajectory in detail. For this purpose, we use a variable J, which is defined as the *performance measure* (Kirk, 1970). We need to also define, whether we want to minimize or maximize the value of J. In this thesis, we always want to maximize it. We then have the option to either calculate the value of J at the end of the trajectory, or increment its value already during trajectory integration. It is common to combine both methods. It certainly makes a difference if the trajectory leads us to where we originally wanted to go (distance from the end of trajectory point to the target point). We also desire as little control inputs during the ascent as possible, since aerodynamic control surface deflections cause increased drag. These are just common examples, many more parameters can be treated the same way.

In some applications, it might be better to just reach a target altitude as efficiently as possible, efficiency being typically measured based on fuel burn. For some missions, it will be desirable to cover as much ground distance as possible, during the ascent. For this purpose, we define J_1, which is our first component of the total performance measure J. J_1 is directly calculated from the ground distance covered during the ascent, and its effect on J is modulated by a weighting coefficient, which we denote ϕ_1.

When we want to maximize the range of the suborbital arc, we need to maximize our final velocity. We now define the component J_2, and the associated weighting coefficient ϕ_2. The final velocity, and the associated coefficient, will directly determine the value of J_2.

Whereas the other components of J are calculated at the end of each trajectory, the J_3 is always calculated during the trajectory trace. Its value is the integral of the time rate of change of α (AOA). As a reminder, α is the angle of attack of the vehicle's reference body line with reference to the air mass, as the vehicle shoots through the atmosphere (Shevell, 1983, p. 55). By setting a good value for the weighting coefficient ϕ_3, we should effectively remove abrupt pitch attitude changes, caused by the optimization algorithm. This should help to ensure smooth and efficient flight for the vehicle. We assign a negative value for the J_3, because our optimization aims to maximize J. Now, we calculate J_3 as

$$-J_3 = \sum_{n=1}^{f} |\alpha(t_n) - \alpha(t_{n-1})|^2, \quad (t_0, t_1, t_n, \ \ldots\ , t_f), \quad n \in \mathcal{Z}, \quad t \in \mathcal{R}, \quad (5.1)$$

where $|\alpha(t_n) - \alpha(t_{n-1})|$ is the angular difference between the value of α at time t_n, and the previous time step. t_0 is the time value at the beginning of the trajectory trace, and t_f at the end.

For the time-based costs, we assign the J_4, its magnitude being simply the elapsed time in seconds, and its sign reversed. Therefore, we write $-J_4 = t_f - t_0$.

We also need to consider the cost of fuel. For that purpose, we assign J_5 to be the vehicle total mass, at the end of a trajectory trace. While aiming to maximize it, we are at the same time maximizing the fuel remaining.

Our last term is J_6, representing the final altitude reached during the pow-

ered ascent. We can now calculate the value of the performance measure J.

$$J = \sum_{i=1}^{6} \phi_i J_i \tag{5.2}$$

We emphasize here, that the ϕ weighting coefficients are an important tool in obtaining useful multiobjective optimization results. Also, in optimization the values given by a cost function should preferably be *positive-definite*. Therefore, the weighting coefficients should be selected so, that the performance measure J can have only positive values.

Now, we have formulated our problem in a clear way. We want to find a trajectory, which will maximize the value of J. We can use the ϕ weighting coefficients to ensure that the trajectory fulfills our specific mission needs.

5.2 Constraints

Many real-world optimization tasks include several constraints, which must all be satisfied for the set of acceptable solutions. Ours is no different, and for that reason, the following constraints were included.

Our first limiting factor, for an acceptable ascent trajectory, is the acceleration limit for human occupants. This we denote as $\lim \dot{V}_{\text{crew}} = 6$ g, where we have selected six times the force of gravity as an arbitrary limit. This constraint was already discussed by Sänger in his study (Sänger and Bredt, 1944).

The obvious second limit is imposed by the structural and material properties of the airframe, and is denoted $\lim \dot{V}_{\text{airframe}}$. Using modern design and construction methods, this limit will always be higher than the crew limit.

The dynamic pressure q, is an important physical quantity, which is derived from the air density and velocity as

$$q = \frac{1}{2}\rho V^2. \tag{5.3}$$

Its constituents were explained in Chapter 3. It imposes a severe loading on the vehicle structure, and therefore a safe limit is always defined by the engineers. In our case, we set an arbitrary limit as $\lim q = 35$ kPa. Based on available literature, a limit of 60 kPa could also be used (Reed et al., 1979)

The limit for the coefficient of lift was already discussed in Chapter 3. Because it is directly derived from AOA during simulation, it is sufficient to only include constraints for the lowest and highest allowable AOA for each time step.

5.3 Simulation Setup

Before we include the optimization algorithm, we need to build a simulation, which is able to do trajectory traces, and calculates the performance measure *J*. Table 5.1 provides an overview of the simulation built for this thesis.

First, we need a starting point for creating the simulation. For a 3-DOF point mass treatment, we start with the Newton's second law

$$\mathbf{F} = m\mathbf{a}, \tag{5.4}$$

and then write it in invariant tensor form, which supports the frames of reference and coordinate systems we have selected. Now, Newton's second law can be written as

$$m\Big[\frac{\mathrm{d}\mathbf{v}_B^L}{\mathrm{d}t}\Big]^L = [T]^{LV}[\mathbf{f}_{\mathrm{a,p}}]^V + m[\mathbf{g}]^L, \tag{5.5}$$

where the left side is mass times acceleration and $\mathbf{f}_\mathrm{a}$ and $\mathbf{f}_\mathrm{p}$ the aerodynamic and propulsive forces acting on the vehicle. The remaining force required for a 3-DOF simulation is the force of gravity, the last term on the right side. These forces and the Newton's second law are sufficient - Euler's law is only required for a 6-DOF simulation and is omitted here (Zipfel, 2000, p. 155). Once again, I'm using the notation presented by Zipfel in his book (Zipfel, 2000, p. 22).

The $[T]^{\mathrm{LV}}$ is a transformation matrix, which is needed to handle transformations between the velocity and local-level coordinate systems (Zipfel, 2000, p. 81). We still need another transformation matrix, the $[T]^{\mathrm{VL}}$, to provide transformations to the opposite direction. These two are sufficient for our 3-DOF simulation.

The point mass treatment also requires that we assume a *rigid body*. This means that we do not consider any elasticity in wings or other parts of the vehicle, which would affect the aerodynamics and C.M.

Item	**Value**	**Description**
Degrees of freedom	3	3-DOF
Equations of motion	The *Cartesian approach*, where the primary state variables are the vehicle's inertial velocity and position components.	
Location of the point mass	Vehicle C.M.	
Coordinate systems used	B or $]^B$ (vehicle body), L or $]^L$ (local level), and V or $]^V$ (velocity)	These are 3D Cartesian coordinate systems
Earth model	Flat earth approximation. Fixed gravity constant. Coriolis effect and transport acceleration not included (Vinh, 1981, p. 51).	
Vehicle drag	$D = f(C_L, Ma)$	GHAME drag polar.
Vehicle thrust	$T = f(H, Ma, \mathbf{p})$	3-stage propulsion system.
Vehicle structure	Rigid body	
Time step for integration	$\Delta t = 1$ s	
Integration method	Newton-Cotes	
State variables	Mass, inertial position, inertial acceleration, inertial velocity, true airspeed, Mach number, ground speed, altitude, ground range, angle of attack, flight path angle and dynamic pressure.	
Initial values for some of the state variables	H = 500 m above mean sea level. Inertial velocity 205 m/s. $\alpha = 22°$. $m_i = 136077$ kg.	At time t_0
Performance measure	Value of J, as explained in Section 5.1.	Optimization
Control vector $\mathbf{u}$	Target true airspeed (TAS) for each time step.	Optimization
Control vector $\mathbf{p}$	Propulsion staging. Mode 1 (turbofan), mode 2 (ramjet) and mode 3 (rocket).	Optimization

Table 5.1: Simulation setup

At the beginning of each time step, we update the atmospheric variables, as a function of altitude. In this thesis, we use the values for the ISA standard atmosphere. However, one could also fetch real-world values from a suitable weather database.

Next, we fetch the target airspeed u_t for this time step. It comes from the control vector $\mathbf{u}$. This speed is a so called *true airspeed* (TAS), which is equal to the inertial speed in zero wind conditions, with reference to $]^{\mathrm{L}}$. We also calculate the Mach number, as a function of TAS and local speed of sound. Dynamic pressure is calculated as a function of air density and TAS.

We proceed to calculate the value of the engine thrust, as a function of engine stage, altitude, Mach number and throttle position. The active engine stage is given by our control vector $\mathbf{p}$. Fuel flow is also calculated, as the mass of fuel consumed per unit time.

At this stage, we know the initial velocity for the time step, available thrust and target airspeed. In order to follow the target airspeed, a *PID controller* was added to the Matlab code (Fortescue et al., 2005, p. 319). This basically simulates an autopilot pitch channel operation. The PID controller is able to reduce AOA, if the speed starts to lag the target speed and vice versa. It is also able to dampen rapid changes, and reduce overshoots, as the controller tries to follow the target speed.

Next, we derive the values for lift and drag, with reference to $]^{\mathrm{V}}$. This is done by first using GHAME aerodynamic data tables and PCHIP interpolation, to fetch values for C_L and C_D. Lift and drag forces are then calculated using equations 3.2 and 3.3.

Then we need to update the two transformation matrices, $[\mathrm{T}]^{\mathrm{VL}}$ and $[\mathrm{T}]^{\mathrm{LV}}$. Forces along each axis of the velocity coordinate system are calculated. With these forces, and the transformation matrix $[\mathrm{T}]^{\mathrm{LV}}$, we calculate the resultant acceleration values for each axis of the local-level coordinate system. Then we update the velocity components, by accounting for the acceleration in unit time and the time step. Based on the velocity, we also update the vehicle position vector $\mathbf{r}$ and the flight path angle γ. This means, that the position and velocity of the C.M. can be directly integrated in the local-level coordinate system. In contrast, the aerodynamic lift and drag forces are given in velocity coordinates.

The mass of the vehicle is reduced by the mass of fuel consumed during

the time step.

At this point, we have handled all the operations for one time step. The software then loops back, and integrates all the time steps until the end of the trajectory.

Zipfel's GHAME3 work can be used as an example of this type of simulation (Zipfel, 2000, p. 157). After the equations of motion are derived, using invariant tensor form, they are expressed as vectors and matrices in computer code.

We end this section with some words about software tools.

All the software for this thesis was created in Matlab. It is a multi-use numerical computing environment, using its own proprietary programming language. Matlab natively supports matrix manipulations and graphical plotting. There are also other useful functions, including standard atmosphere models and associated data. Matlab has dedicated add-on modules for aerospace projects, however those were not utilized in this thesis.

DATCOM was developed for the USAF by McDonnell Douglas in 1970s. It is a modular software that allows cost effective preliminary design evaluations to be conducted for aerospace vehicle concepts. The software is able to produce a wealth of useful information (Williams and Vukelich, 1981). It has been used in several aerospace projects. Open source code from project OpenDatcom is also available.

CADAC was initially developed in 1966 by Litton Industries, and it has been further developed ever since. It has been used in numerous aerospace projects in US and Great Britain. Zipfel has promoted this software in many of his publications. Originally developed in FORTRAN, it has been gradually transformed to C++ (Zipfel, 2011).

5.4 Implementation of the Simulated Annealing

Simulated annealing (SA) is a technique, which can be implemented in many ways, depending on the phenomenon under study. For a certain optimization task, there might be several feasible ways to implement SA. The first thing to point out, regarding the implementation of SA for this thesis, is that whereas SA is often used to minimize the *energy*, we are using it to maximize

our *performance measure* J. A simple sign reversion will suffice here.

Because the trajectory tracing is already computationally intensive, we should limit the number of objects that we desire to variate with SA. Therefore, a small subset of objects was used to produce two control vectors, $\mathbf{u}$ and $\mathbf{p}$. This small subset of objects formed a $m \times n$ matrix $\mathbf{U}$. Details of this matrix are found in Chapter 6. During the SA optimization, we then variate values in $\mathbf{U}$ between each iteration. An overview of this process is shown in Figure 5.1.

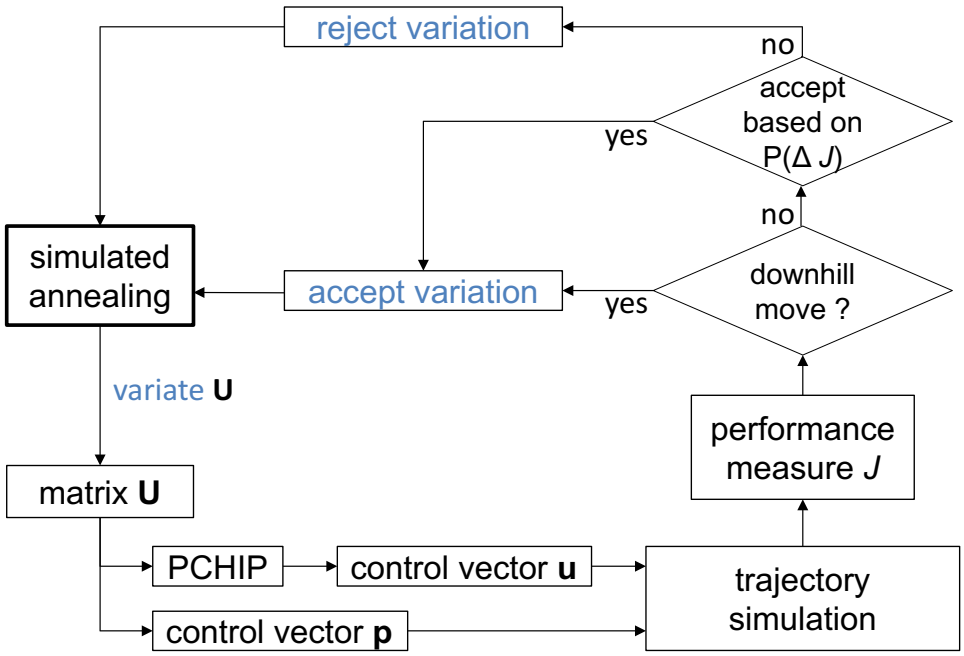

Figure 5.1: Flowchart illustrating how SA optimization was integrated into the trajectory simulation. In this simplified diagram, we can start by having SA variate $\mathbf{U}$. The two control vectors are then constructed and the trajectory simulation traces a complete trajectory from t_0 to t_f. We then calculate J for the trajectory. If the value of J has improved, in our case increased, we accept the move and go to the next iteration. In other case, we still have an option of accepting an *uphill move* based on SA probability function. After each iteration, the SA *temperature* is decreased. This process continues until a predefined number of iterations have been completed.

The control vector **u** is for target true airspeed (TAS), and the control vector **p** controls engine staging. Our choice of the SA *move class* allows only discrete mutations, for the array values in **u** and **p**.

$$\mathbf{u} = [u_1 \quad u_2 \quad ... \quad u_n], \quad u_n, n \in \mathbb{Z} \tag{5.6}$$

$$\mathbf{p} = [p_1 \quad p_2], \quad p_1, p_2 \in \mathbb{Z} \tag{5.7}$$

Because the simulation will reject trajectories that do not fulfill the constraints, as later explained in Chapter 6, we conclude that the set of acceptable trajectories forms the SA *state space*.

Figure 5.2 shows, how the SA was setup to variate **U**. During each iteration, we first randomly select, how many adjacent array values we change at that instance. Allowed choices were one (a.), two (c.) or three (b.) adjacent array values to be moved during the same iteration. Next, we calculate the direction and magnitude of the move, in this case speed increment or decrement of some meters per second. The magnitude of the change is random, with the range of possible values dictated by the SA *temperature* variable. At higher temperatures, larger variations are possible.

We have not yet produced the complete control vector **u** required for the trajectory trace. To build **u**, we use a technique called Piecewise Cubic Hermite Interpolating Polynomial (PCHIP). By applying PCHIP to the matrix **U**, we obtain a complete control vector **u**, which is sufficiently smooth for our purposes. After all, rapid changes in the target speed during flight are not desirable. Example of this method is given in Figure 5.3. The last two rows in **U** were reserved for the engine staging control. The control vector **p** was directly determined by the values contained in the last two rows of **U**.

SA belongs to a group of algorithms, that can accept a move, even if the performance measure decreases. Using this principle, SA avoids being trapped in a *local minima*. The acceptance of moves, that degrade the performance measure temporarily, is handled based on probability. In the Metropolis algorithm (Kirkpatrick et al., 1983, p. 672), the probability is calculated as

$$P(\Delta E) = e^{\frac{\Delta E}{k_B T}},$$

which is the Equation 4.4 covered in Chapter 4. When implementing SA to optimization problems, which are not based on a direct physical model, the temperature variable T can be assigned any arbitrary values. This enables

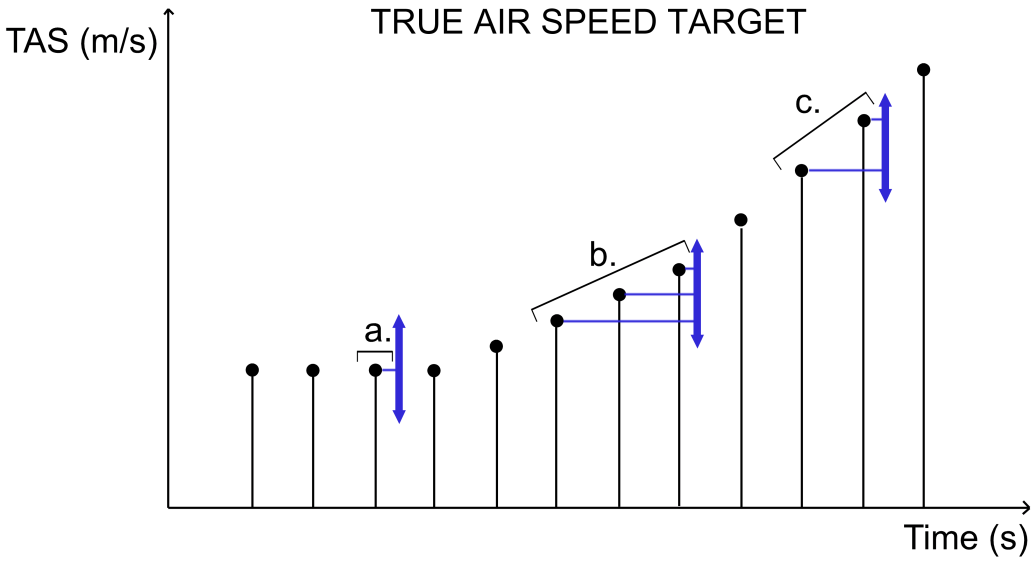

Figure 5.2: The method used to implement simulated annealing. During each time step, one (a.), two (c.) or three (b.) adjacent array values were randomly selected for variation. The direction and magnitude of the move is random, with the range of possible values dictated by the SA *temperature* variable. At higher temperatures, larger variations are possible.

us to choose T, so that we can eliminate the Boltzmann constant. Also, we do not need to study the energy E of the system, instead we can replace it with the performance measure J, which we are aiming to optimize. Thus, we can derive the probability as

$$P(\Delta J) = e^{\frac{\Delta J}{T}}, \tag{5.8}$$

where ΔJ is the change in the value of the performance measure, resulting from the move. The Equation 5.8 is used, in this thesis, to calculate the acceptance probability of degrading moves. Table 5.2 show examples of $P(\Delta J)$ for different values of T and ΔJ. As the "temperature" cools down, the probability of accepting a move that degrades the performance measure reduces.

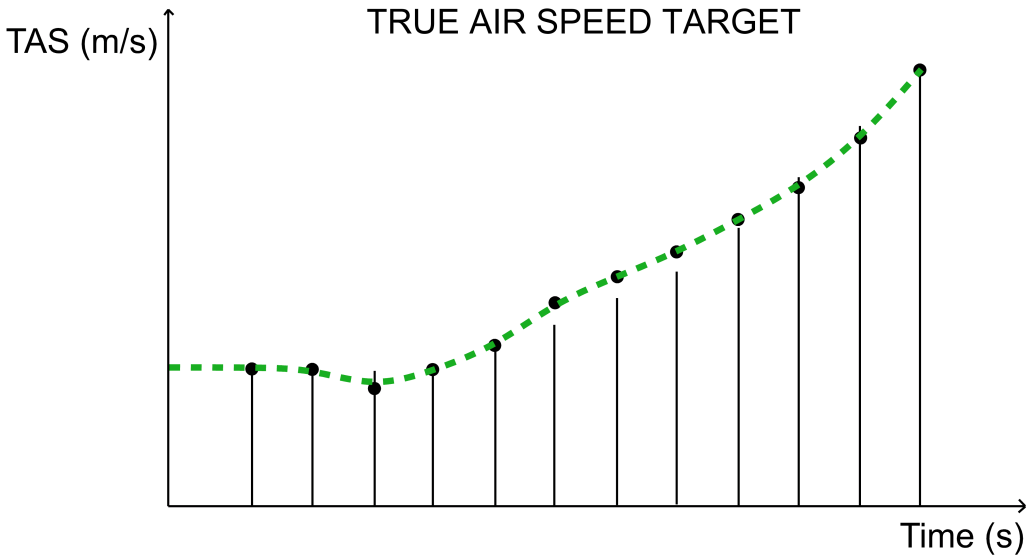

Figure 5.3: Example of smoothing and filling using PCHIP. If we were to simply connect the points given by the matrix **U** (black points) with lines, there would be sharp speed target changes. This would cause undesirable rapid AOA changes during flight. PCHIP overcomes this issue by providing a smooth and complete control vector **u** (dashed green line).

$\boldsymbol{\Delta J}$	$\boldsymbol{T}$	$\boldsymbol{P(\Delta J)}$
-50	50	0.3679
-40	15	0.0695
-40	1	$4.24835 \cdot 10^{-18}$

Table 5.2: Example values for $P(\Delta J)$. As the "temperature" cools down, the probability of accepting an uphill move reduces.

Chapter 6

Results from the Simulation Runs

In this chapter, we lay out the results from the trajectory trace simulations. We see the baseline speed profile, and the resulting baseline trajectory, which are then used as a starting point for the SA optimization runs.

6.1 The Baseline Trajectory

Before proceeding into the task of optimizing an ascent trajectory, we should have a baseline which acts as a reference and a starting point. For this purpose, the control vector $\mathbf{u}$ was initially populated with arbitrary baseline values. The length of this vector was preset to 1441 cells, where each cell represents a one second time step for the simulation. This limited the duration of each trajectory simulation to a maximum of 1441 seconds. To populate the cells in $\mathbf{u}$, and to enable the SA algorithm to variate $\mathbf{u}$ during the annealing period, a matrix named $\mathbf{U}$ was defined. Size of this $m \times n$ matrix was set to 45×2. The values for the first column of $\mathbf{U}$ were then set using a simple for-loop starting from 1 and incrementing each subsequent row value by 45. The first column now represented simulation time in 45 second intervals. The second column was also populated by for-looping, having linearly increasing target speed values for each row, with different slope values for the turbojet and the ramjet stages. The values for the rocket propulsion phase were set to increase non-linearly. The subsequent matrix is plotted in Figure 6.1, up to the time step of 1200 s, because as we will later see, the simulation runs do not take longer in this study. Now, $\mathbf{u}$ was created from $\mathbf{U}$ by using PCHIP, as mentioned in Chapter 5. See Figure 6.2 for the resulting target speed profile, and Figure 6.3 for the corresponding baseline trajectory.

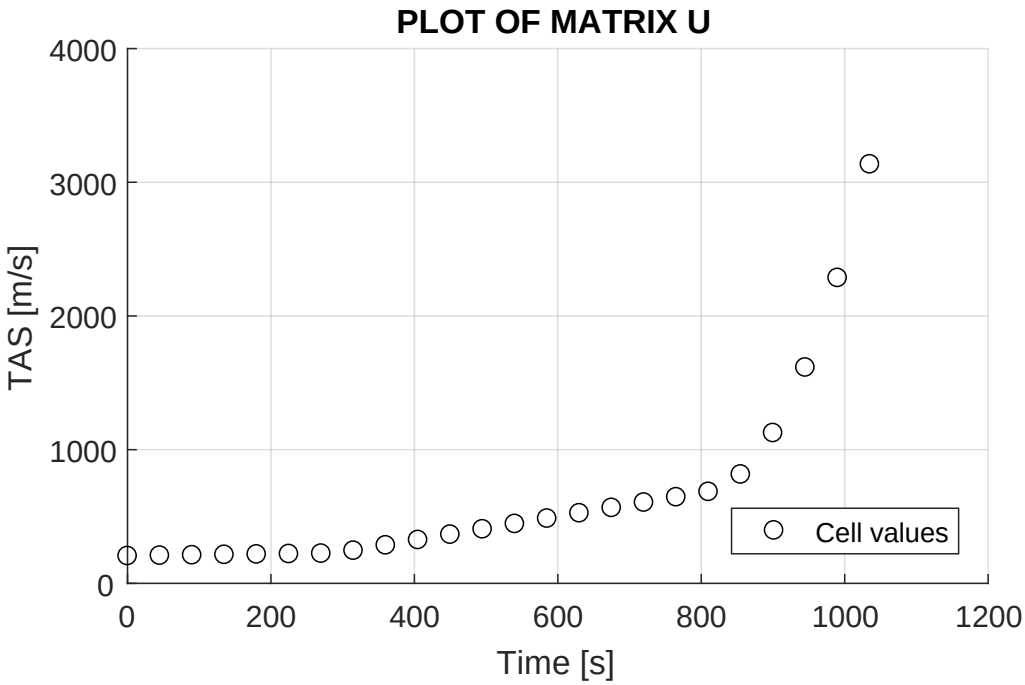

Figure 6.1: Plot of the matrix **U**. X-axis represents the first column of **U**, and Y-axis represents the second column. This plot shows the values for the baseline trajectory. During optimization, SA variates each cell value to a higher or lower true air speed target. After each variation, the complete control vector **u** is then smoothly interpolated from **U** by using PCHIP.

The Matlab code created as part of this thesis included a PID controller operated autopilot, which had only one channel (pitch control). The simulated vehicle then used the autopilot, to lower or raise the nose of the vehicle during flight, to maintain the target airspeed contained in the control vector **u**. We can see from Figure 6.2, that it was able to follow the airspeed target quite accurately for the baseline trajectory. Later we will see, how the autopilot performs when SA variates **u** and **p** (via **U**). It is also worth to mention, that the baseline trajectory was calculated assuming zero wind at all altitudes. The atmospheric temperature profile followed the ISA standard atmosphere model.

Figure 6.4 shows the dynamic pressure q plot for the baseline trajectory. The Matlab code was set to reject any trajectories, which exceeded the 35 kPa dynamic pressure limit. This feature was used in the SA optimization runs, to discard unsafe trajectories, which would have exceeded vehicle structural

limits. Another baseline plot is for the thrust (T) and drag (D) (Figure 6.5). The thrust and drag plot is shown here as an example. Many similar type of plots can be made using other state variables. These plots are useful in a trajectory post-analysis.

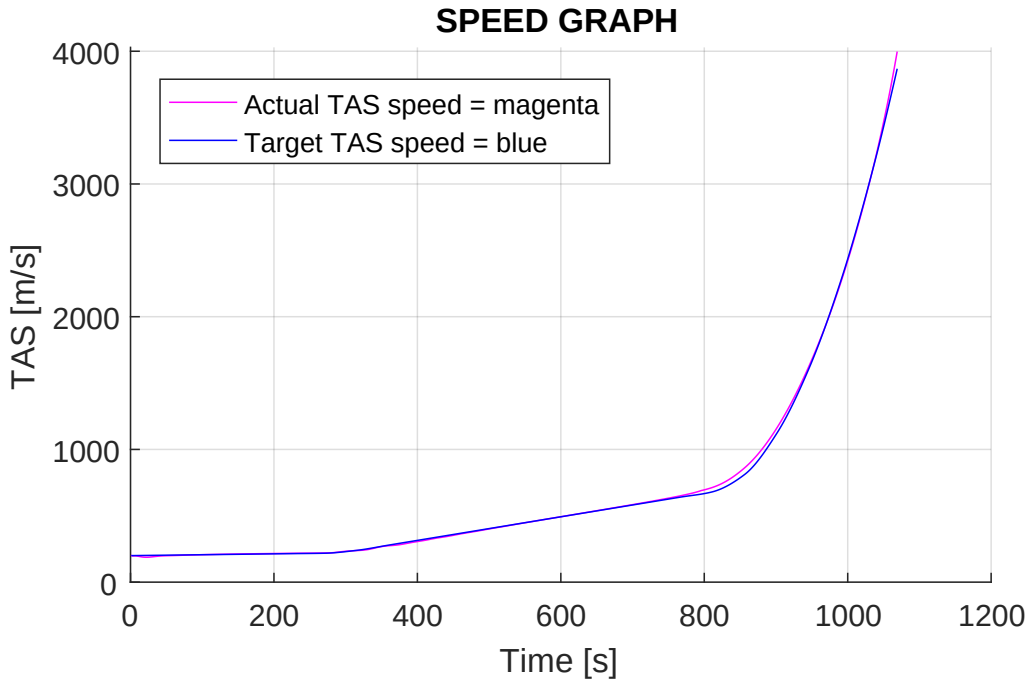

Figure 6.2: The baseline control vector $\mathbf{u}$, and the resulting actual speed. X-axis is the time since t_0, and Y-axis shows corresponding true air speed in relation to the airmass. The plot also shows, that the PID controller operated autopilot pitch channel could follow the target speed quite accurately.

The initial values for the control vector $\mathbf{p}$ were set as follows.

$$\mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} 8000 \\ 25000 \end{bmatrix} \tag{6.1}$$

Both values represent altitudes in meters, and were used as triggers for the engine staging. In the baseline trajectory, switching from turbojet to ramjet took place at 8 km altitude. Switching from ramjet to rocket engine was done when reaching 25 km altitude.

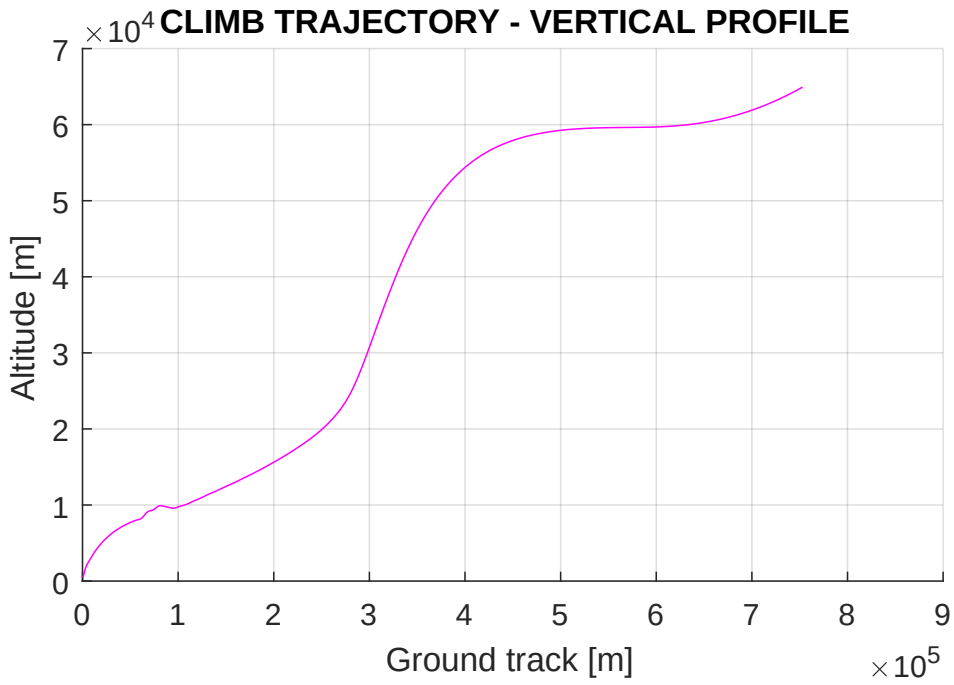

Figure 6.3: The baseline trajectory, as it would be seen by an observer located South of the launch site. X-axis represents the ground distance that the vehicle has covered. Y-axis is for the flight altitude.

Chapter 5 introduced the weighting coefficients, for calculating the value of the performance measure J. Table 6.1 shows the values used for the baseline trajectory, and the first optimization runs.

Coefficient	**Value**	**Factors**	**Description**
ϕ_1	0.001	J_1	Ground distance
ϕ_2	0.5	J_2	Final velocity
ϕ_3	90000	J_3	Rate of change of AOA integrator
ϕ_4	0.8	J_4	Time elapsed
ϕ_5	0.05	J_5	Fuel remaining
ϕ_6	0	J_6	Final altitude

Table 6.1: Performance measure components and weighting coefficients. For the baseline trajectory, these had no effect on the flight path, instead they were used at the end of the trajectory to calculate value for the baseline performance measure J.

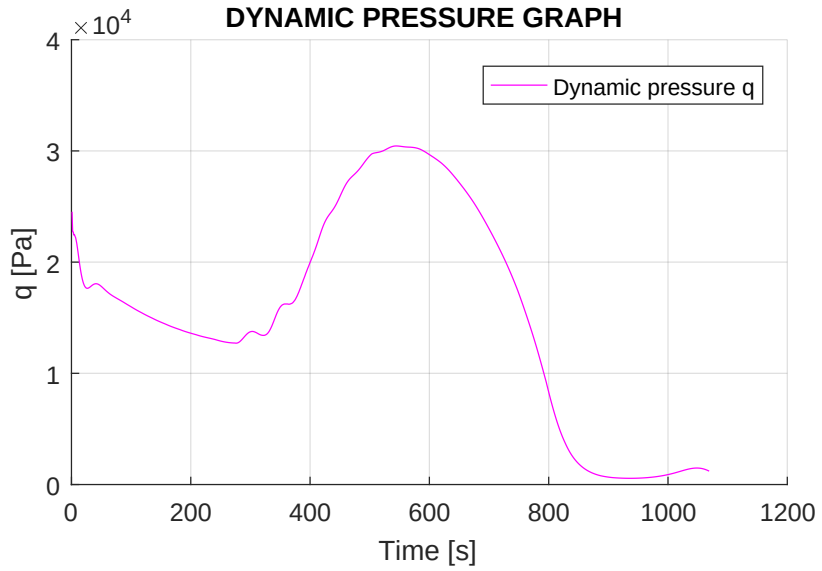

Figure 6.4: Dynamic pressure plot for the baseline trajectory. This shows, that one of the constraints, $\lim q = 35$ kPa, was respected for the baseline trajectory. During optimization, any trajectory exceeding this limit was automatically discarded.

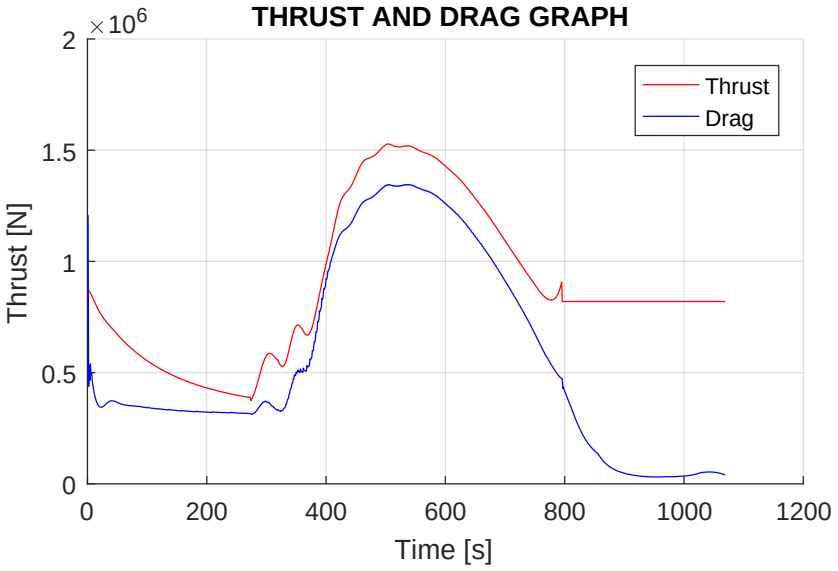

Figure 6.5: Engine thrust (T) and aerodynamic drag force (D) for the baseline trajectory. Switch from turbofan to ramjet takes place at around t = 280 s, ramjet to rocket at around t = 800 s.

6.2 The Optimized Trajectory for Conservation of Fuel

The weighting coefficients values shown in Table 6.1, were selected so, that the conservation of fuel had a prime impact on the performance measure J. Using the same simulation setup, as for the baseline trajectory, SA looping was then enabled in the Matlab code for 50,000 trajectory trace iterations. Six optimization runs were then executed, three of which used a linear cooling scheme, and the other three using exponential cooling. In both cases, the initial value for the temperature variable was set to 50. Trajectory traces were calculated up to 65 km altitude, for the baseline and optimized cases.

For the linear cooling schema, the temperature was decreased after each iteration, so as to arrive at zero temperature during the last iteration. For the exponential cooling scheme, temperature for each iteration was calculated as

$$T = 0.9999^k, \tag{6.2}$$

where k is the step-by-step increasing iteration number. By using this empirically derived schema, the annealing temperature had reduced to 0.3368 for the last iteration.

After the six runs, the resulting optimized trajectories were manually inspected for feasibility. The results can be found in Table 6.2. It is evident, that the robustness of SA using this setup is not very good. Two out of six trajectories had to be discarded, as they did not possess any desirable qualities. In the next chapter, we discuss about possible remedies to this issue. Figure 6.6 shows the baseline trajectory, and the optimized trajectories.

Cooling schema	**Simulation run**	**Result of manual inspection**
Exponential	Exp1	Accepted
Exponential	Exp2	Accepted
Exponential	Exp3	Discarded based on the low value of J
Linear	Lin1	Accepted
Linear	Lin2	Discarded based on the low value of J
Linear	Lin3	Accepted

Table 6.2: Manual inspection and filtering of resulting trajectories.

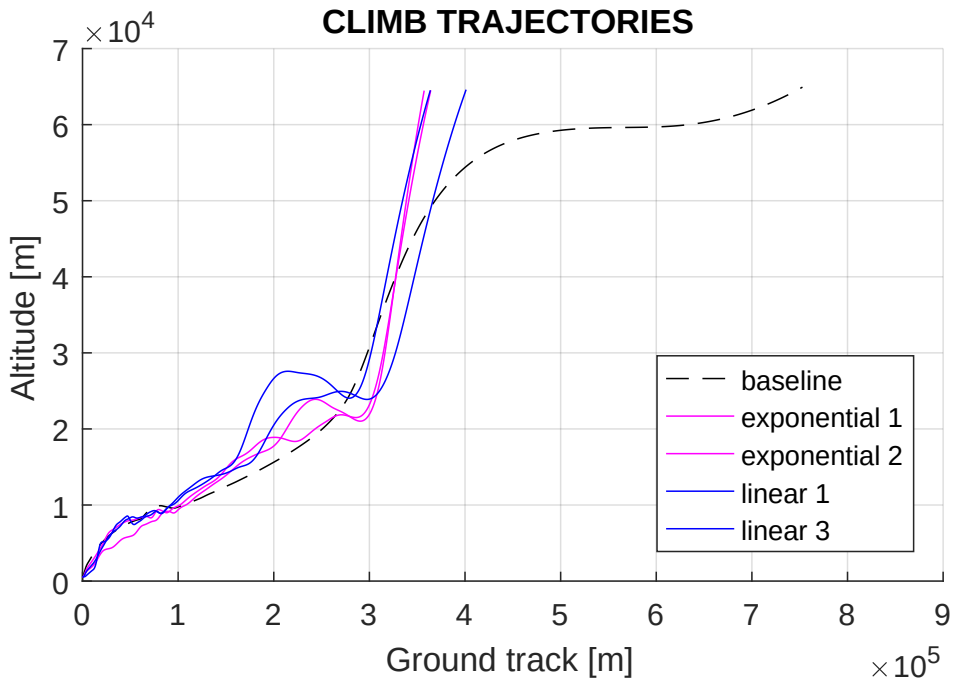

Figure 6.6: The baseline trajectory and the optimized trajectories, as they would be seen by an observer located South of the launch site. X-axis represents the ground distance that the vehicle has covered. Y-axis is for the flight altitude. All four optimized trajectories had an improved value of the performance measure J.

Results for the accepted trajectories are shown in Table 6.3. Note that the values of J do not represent any physical quantity. The range of values for J is determined by the arbitrarily chosen values for the weighting coefficients. For all the experiments in this thesis, the weighting coefficients were derived empirically.

In order to analyze the operation of SA, we can monitor the value of J during an optimization run. An example is given in Figure 6.7. Ideally, SA first finds optimality for the gross features, and later focuses on the fine details, as the temperature cools down (Kirkpatrick et al., 1983).

Figures 6.8, 6.9 6.10 and 6.11 show the control vector $\mathbf{u}$, after the last iteration, for each of the simulation runs. The baseline is shown as a black line. The actual speed is the result of the vehicle trying to follow the assigned speed target, as closely as possible, guided by the autopilot.

Run	**Elapsed time (s)**	p_1 **(km)**	p_2 **(km)**	**Final velocity (km/s)**	**Fuel remaining (kg)**	ΔJ
Baseline	1069	8	25	4.0	1,886	0
Exp1	848	8.5	30	1.1	63,674	6447
Exp2	844	7.8	30	1.0	64,968	6529
Lin1	818	7.0	30	1.3	58,475	6385
Lin3	812	8.5	30	1.2	63,669	6556

Table 6.3: Results of the optimization runs and the baseline for comparison. Fuel mass at the beginning of each trajectory trace was 115,670 kg, and the resulting total mass of the vehicle 136,077 kg. All four optimized trajectories had an improved value of the performance measure J.

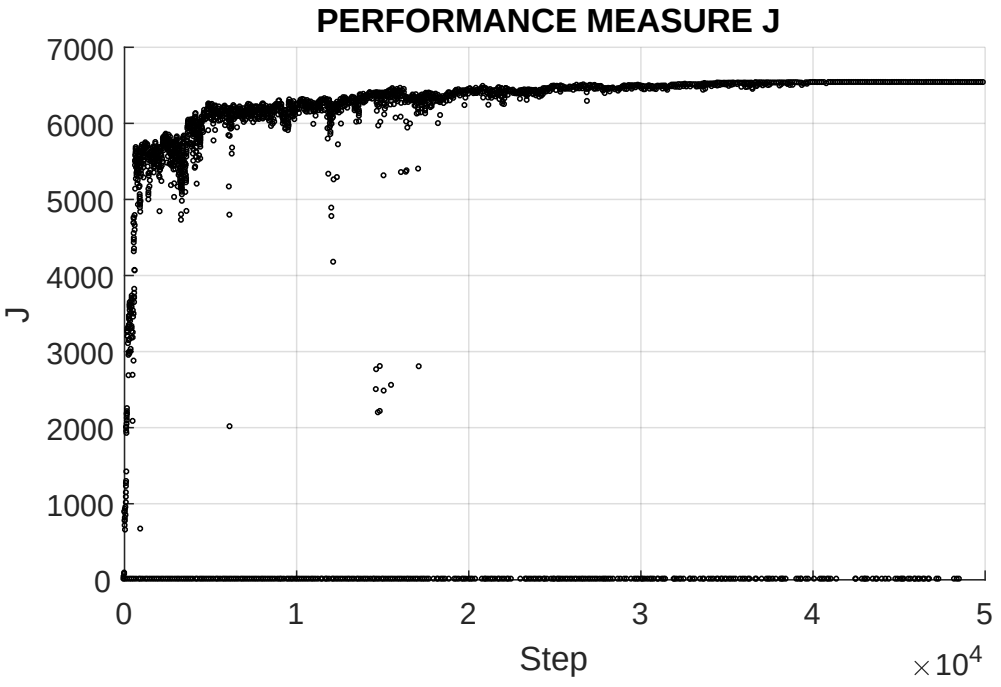

Figure 6.7: Value of the performance measure J versus iteration step for the Exp2 run. The optimization run starts from the left. There are certain steps, where the value of J jumps up. These are typically the points, where the gross features of the system rearrange to an improved state σ. The individual dots, below the equilibrium value of J, are results from uphill moves accepted based on the probability function. As the system cools down, the occurrence of these uphill moves reduces significantly, an inherent feature of SA.

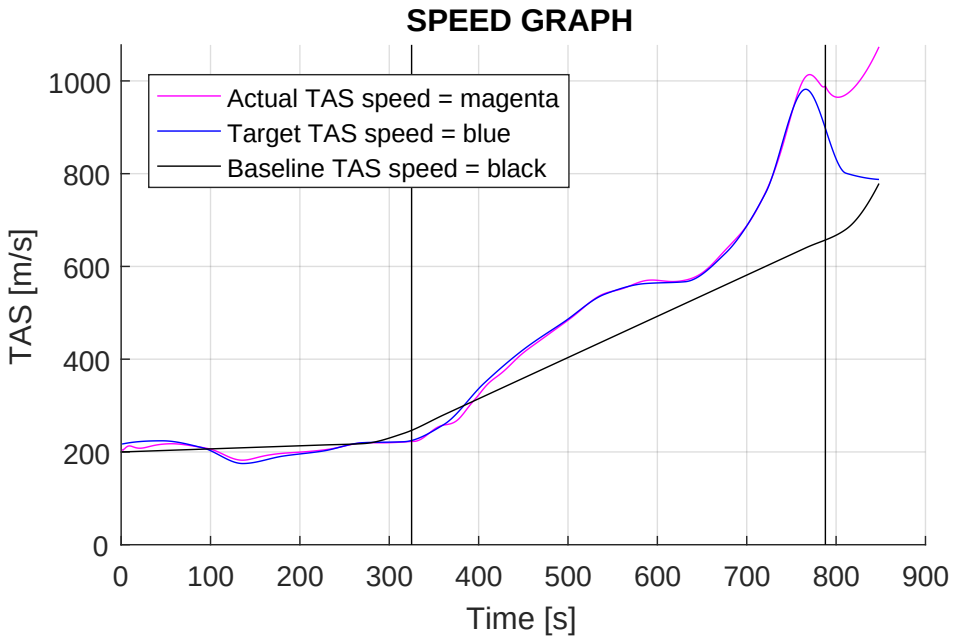

Figure 6.8: Resulting control vector **u** for the Exp1 run.

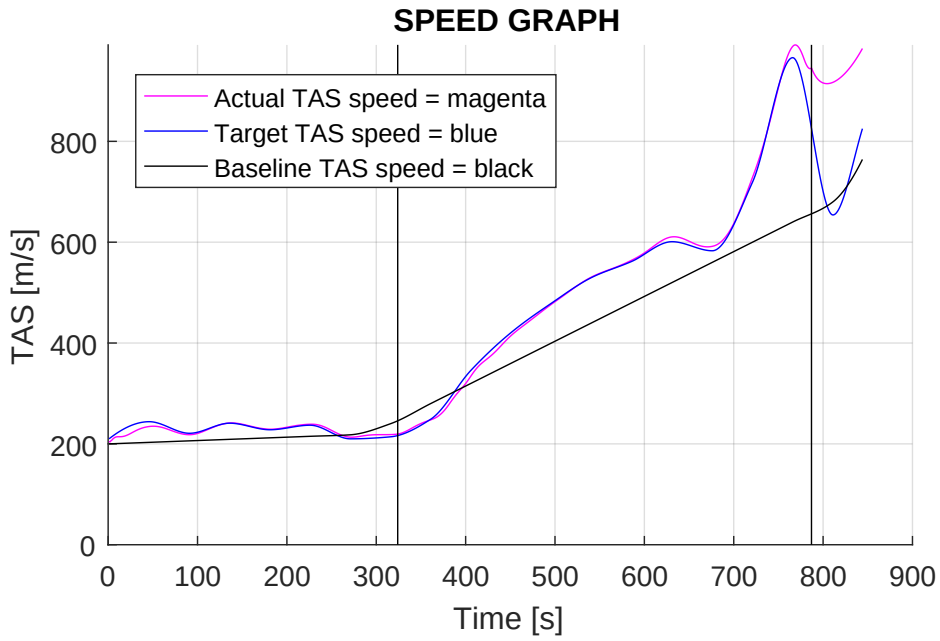

Figure 6.9: Resulting control vector **u** for the Exp2 run.

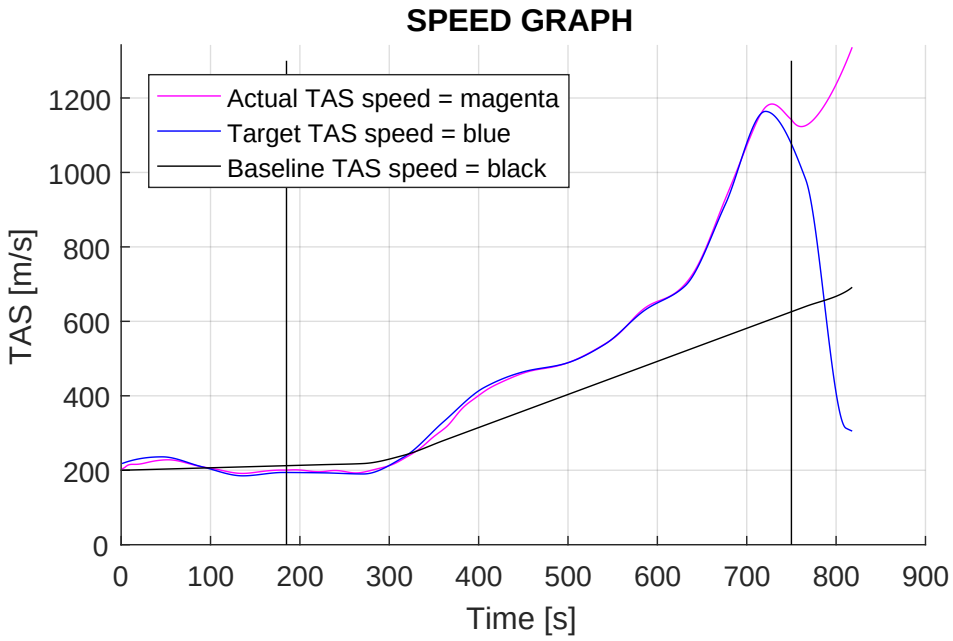

Figure 6.10: Resulting control vector **u** for the Lin1 run.

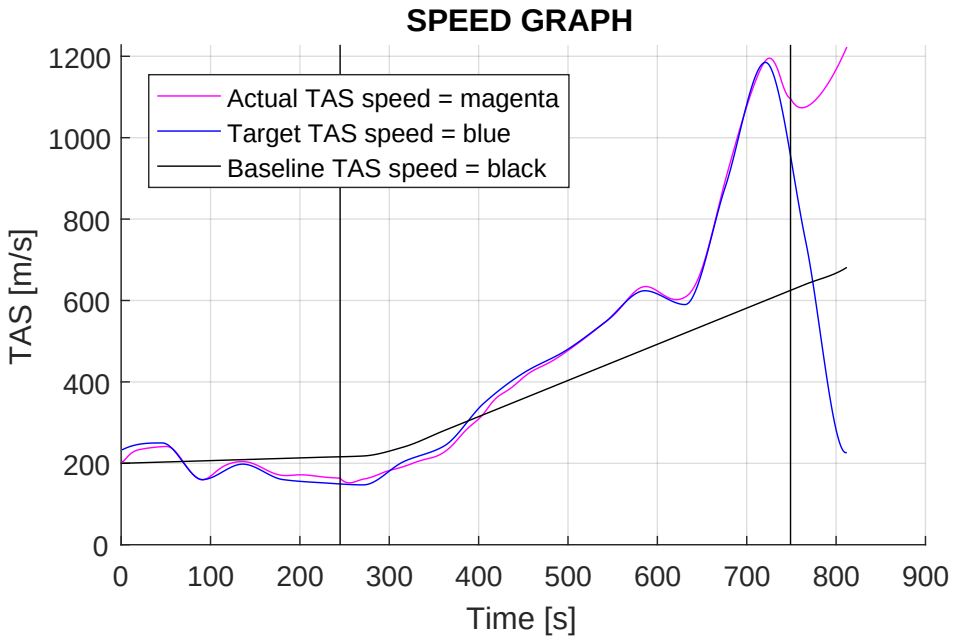

Figure 6.11: Resulting control vector **u** for the Lin3 run.

6.3 The Optimized Trajectory for Maximum Energy State

To find out how flexible SA is for performing different optimization tasks, the weighting coefficients were changed to prioritize final energy state (Table 6.4). The ascent trajectory energy state analysis was explained in Chapter 4. It is the sum of the kinetic (inertial velocity) and potential (altitude) energies of the vehicle. This time, the thrust of the rocket engine was reduced by 30%, and the baseline speed profile was modified accordingly. This was done to allow the autopilot to better follow the speed target at the end of trajectories. Trajectory runs were now traced up to 105 km altitude. The baseline trajectory was not able to reach this altitude, so it was traced up to a point, where all the fuel had been consumed.

In addition to the baseline run, three optimized trajectories were produced, with the SA using linear cooling schema. This time the exponential cooling schema was not used at all. The initial SA temperature was set to 30, and the temperature decreased linearly to reach zero at the last iteration. Initial temperature was reduced, in order to suppress the range of variations produced by SA, as they sometimes seemed to be too large. Number of iterations was reduced to 20,000. This reduced the calculation time significantly, being approximately 5 minutes per optimization run on a desktop PC. Figure 6.14 shows a typical evolution of the performance measure J during 20,000 iterations.

Table 6.5 shows the results from the optimization runs. This time, all three optimization runs produced a feasible trajectory. In all the cases, including the baseline, all the fuel was consumed during the ascent. Each of the optimized trajectories, shown in Figure 6.12, enabled the vehicle to reach a much higher altitude when compared against the baseline. Since the final velocities were close to the value given by the baseline, the total value of the performance measure J was better in all the optimized cases. The small difference in the value of J for Lin5 and Lin6 runs comes from the J_3 term, indicating that the Lin6 run had less fluctuation in the AOA during ascent. As an example, Figure 6.13 shows the control vector $\mathbf{u}$ after the Lin5 optimization.

We can see, that SA was able to produce the desired effect also in this experiment. There is no proof, that any of these trajectories is close to the true optimal trajectory. We can simply state, that the SA algorithm was

given the baseline control vectors **u** and **p**, and it was able to variate them in a way, that the final energy state of the resulting trajectories was increased, while also ensuring that all the constraints were respected.

Coefficient	Value	Factors	Description
ϕ_1	0	J_1	Ground distance
ϕ_2	4	J_2	Final velocity
ϕ_3	175000	J_3	Rate of change of AOA integrator
ϕ_4	0	J_4	Time elapsed
ϕ_5	0	J_5	Fuel remaining
ϕ_6	0.2	J_6	Final altitude

Table 6.4: Performance measure components and weighting coefficients. Note that the final altitude is for the ascent trajectory tracing, not for the resulting ballistic arc.

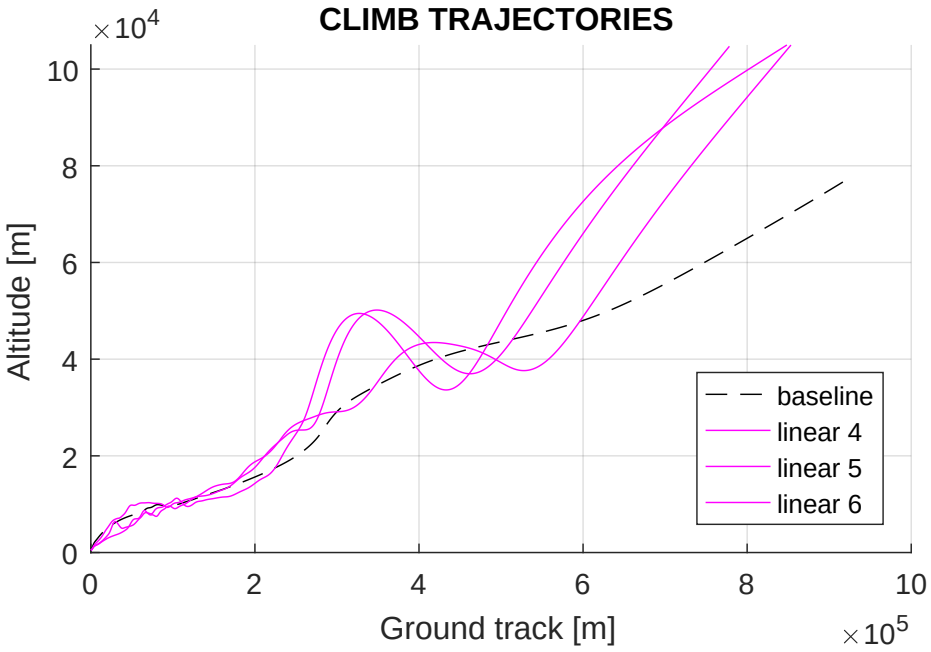

Figure 6.12: The baseline trajectory and the optimized trajectories, as they would be seen by an observer located South of the launch site. X-axis represents the ground distance that the vehicle has covered. Y-axis is for the flight altitude. All three optimized trajectories had an improved value of the performance measure J.

Run	**Elapsed time (s)**	p_1 **(km)**	p_2 **(km)**	**Final velocity (km/s)**	**Final altitude (km)**	ΔJ
Baseline	1195	8	25	3.7	77	0
Lin4	1161	7	28	3.6	105	15703
Lin5	1138	7	23	3.5	105	14343
Lin6	1090	7	24	3.4	105	14449

Table 6.5: Results of the optimization runs and the baseline for comparison. In all the three cases, SA was able to improve the performance measure J.

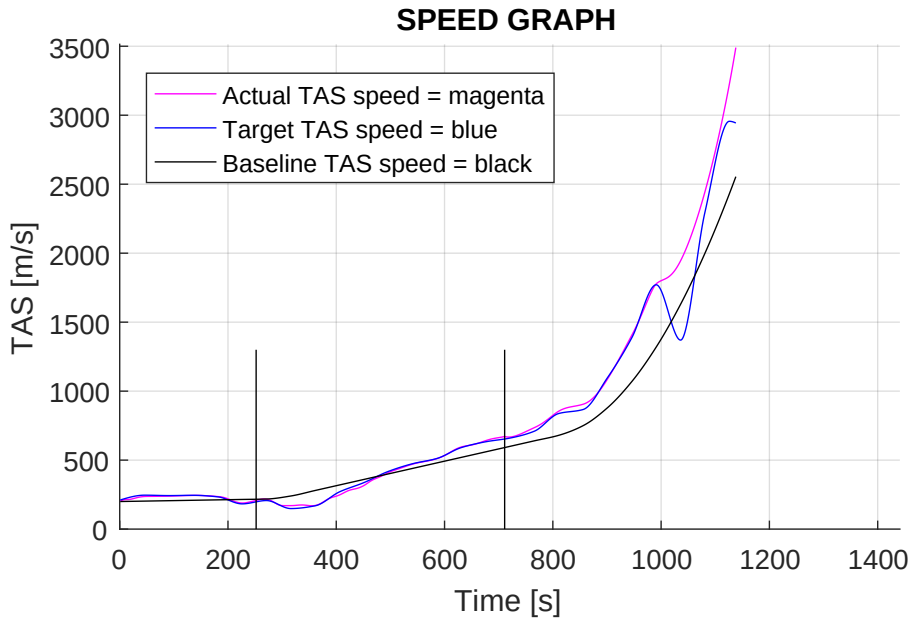

Figure 6.13: Example of an optimized control vector $\mathbf{u}$ for the optimized trajectory Lin5.

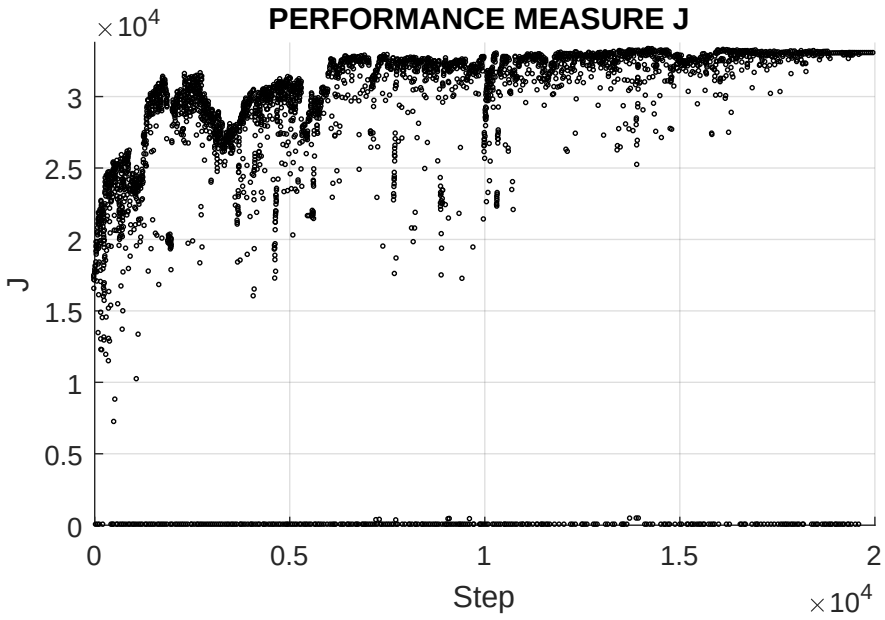

Figure 6.14: Value of the performance measure J, versus iteration step for the Lin4 run. The optimization run starts from the left. There are certain steps, where the value of J jumps up. These are typically the points, where the gross features of the system rearrange to an improved state σ. The individual dots, below the equilibrium value of J, are results from uphill moves accepted based on the probability function. As the system cools down, the occurrence of these uphill moves reduces significantly, an inherent feature of SA.

Chapter 7

Discussion

Here, we look more closely at some details found in Chapter 6, and discuss the findings if any. We also evaluate the performance of SA, based on the results obtained.

7.1 Analysis of the Control Vectors for the Conservation of Fuel

It is interesting to see, how the control vector **u** was transformed by SA. By looking at Figures 6.8, 6.9, 6.10 and 6.11, we can make some observations. First of all, all four optimized speed profiles are similar. The only exception is in the middle of the turbofan powered climb phase. Here, during the Exp2 run, the value of the speed target was higher than in the baseline. For the other three runs, the speed target was always lower for the same section of the control vector **u**. There is no easy explanation for this. Next, we can study the speed profiles around time step $t = 400$ s. We see that in each case, the SA has produced a speed profile which accelerates rapidly from subsonic speeds to approximately Mach 1.7. There is a good reason for this. The total drag of the vehicle has a distinct peak around the transonic speed region (Figure 3.5). The best strategy is to spend as little time as possible in this speed regime, and that's exactly what SA has provided here. The remaining part of the optimized speed profile seems to suggest, that in the final moments of ramjet powered flight, it is beneficial to accelerate as much as possible. This way we benefit from the much better specific impulse I_{sp} of the ramjet, before switching to the fuel-guzzling rocket mode.

The last part of the control vector **u** has little effect, since the rocket powered phase is characterized by maximum angle of attack. Because of this, the au-

topilot ignores any speed targets below the actual speed and the variations in the control vector **u** in this section can have erratic values. This behavior was noticed, when building the trajectory trace simulation for the conservation of fuel case. The trajectories are still valid, the only shortcoming was that the optimization had no effect for the final sections of the rocket powered phase, because the autopilot always drove the angle of attack to the upper limit. The changes that were made for the maximum final energy case provided some remedy to this issue.

When looking at the propulsion mode staging, and the values of the associated control vector **p**, it seems that the choice of altitude for switching from turbojet to ramjet has little effect. At least we don't notice any tendency by SA, to drive this value to any particular extrema. Same is not true for the ramjet to rocket mode switching. In all four runs, the SA drove the switching altitude to as high as possible. The switching altitude was restricted to 30 km in the Matlab code, and that was the final value obtained in all the optimization runs. This can be explained by the much higher specific impulse I_{sp} of the ramjet, when compared to the rocket mode. It is beneficial to use the ramjet as long as possible, for conservation of fuel.

7.2 Analysis of the Control Vectors for the Maximum Energy State

In this case, the robustness of the SA technique used was better, since no trajectories had to be manually rejected. All the three optimized **u** control vectors were similar. The one for the Lin5 run is shown in Figure 6.13. Unlike in the conservation of fuel case, the control vector **p** showed tendency to switch to ramjet mode as early as possible. No clear tendency was observed for the ramjet to rocket switching.

All the optimized trajectories included a short diving phase for fast acceleration, which is often a beneficial technique for high performance vehicles. This was explained in Section 4.1.

7.3 Other Remarks

Selection of linear or exponential cooling schema did not have any noticeable effect on the results. Several more runs should have been conducted, to derive statistically meaningful results on this matter.

In this thesis, the generic simulated annealing method was implemented as part of the Matlab code, which also contained the 3-DOF trajectory simulation. The parameter values for SA were derived empirically, and it is unknown if they are close to the best values for this type of optimization.

Chapter 8

Conclusions

In the beginning of this thesis, we set a goal of finding an answer to the following research question.

Can we find a way to use simulated annealing, to successfully optimize an ascent trajectory of a suborbital spaceplane?

We then looked at the history of suborbital flight and trajectory optimization. Then, simulation and modeling of atmospheric flight was briefly explained. Before we built the required simulation, we familiarized ourselves with optimal control theory, and the simulated annealing (SA) technique. The experimental part included two test scenarios. Several simulation and optimization runs was completed, using the simulation and SA optimization engine explained in Chapter 5. The results and discussion followed. Now, here are the conclusions.

The performance of SA for this type of trajectory optimization looks promising, but more testing and refinements are needed, before it could be developed into a robust and accurate every day tool. The author expects demand for such tools to increase, as suborbital flights become more common (Le Goff and Moreau, 2013). The challenge for all suborbital flights using present day propulsion technologies is, that the *propellant mass fraction* is very high, meaning that most of the takeoff weight consists of fuel. Optimization of the ascent trajectory is important, to save fuel or time, and to maximize useful payload and range.

Many feel, that the airbreathing hypersonic flight is the last frontier of air vehicle design (Anderson, 2007, Chapter 14). For future suborbital vehicles, hypersonic scramjet engines are expected to enable more economically viable

vehicle concepts, by reducing the propellant mass fraction. Still, trajectory optimization will be beneficial for these vehicles.

Direct methods for ascent trajectory optimization are inherently more accurate, when compared to indirect (analytical) solutions (Fortescue et al., 2005, p. 214). The direct methods are used with an up-to-date weather database, giving accurate forecasted atmospheric values throughout the trajectory simulation. Ascent trajectory optimization using SA could be a viable option, among other direct methods listed by Betts (Betts, 1998). Launch window availability should increase, since the planned trajectory can be optimized and simulated with higher precision, resulting from the extremely large state space that SA allows for the associated control vectors. This in turn, enables the use of a weather database having a good spatial resolution, further improving optimization results.

The method, which was used to implement SA in this thesis, enables multiobjective optimization. For the two experiments in Chapter 6, SA was able to produce the desired results. In the first experiment, we were able to significantly increase the remaining fuel mass, at the end of an ascent trajectory. The second experiment similarly showed improved trajectories, when the goal was set to maximize the sum of kinetic and potential energies of the vehicle, at the end of an ascent trajectory.

If the trajectory optimization with SA is implemented in a robust manner, the resulting trajectory represents the *global optimum*, or at least a trajectory which is very close to it. This will then ensure that a flight is able to maximize its useful payload, range or schedule, based on what parameter values are given to the SA algorithm before optimization.

Future work in this area would be to study, how ASA (Adaptive Simulated Annealing) could be implemented for this type of optimization problem. Also, we could study the benefits of including more than two control vectors into the optimization.

Bibliography

Anderson, J. (2007), *Fundamentals of aerodynamics*, 4th edn, Tata McGraw-Hill Education, New York.

Ardema, M., Windhorst, R. and Phillips, J. (1998), Optimization of supersonic transport trajectories, *in* 'Guidance, Navigation, and Control Conference and Exhibit', American Institute of Aeronautics and Astronautics, 4220.

Betts, J. (1998), 'Survey of numerical methods for trajectory optimization', *Journal of guidance, control, and dynamics* **21**(2), 193–207.

Betts, J. (2001), *Practical methods for optimal control using nonlinear programming*, Society for Industrial and Applied Mathematics, Philadelphia.

Cobleigh, B. (1998), Development of the X-33 aerodynamic uncertainty model, Technical report, NASA.

Conway, E. (2005), *High-speed dreams: NASA and the technopolitics of supersonic transportation, 1945-1999*, Johns Hopkins University Press, Baltimore.

Cornelisse, J., Schoyer, H. and Wakker, K. (1979), *Rocket propulsion and spaceflight dynamics*, Pitman, London.

Dietz, R., Arndt, G., Seyl, J., Leopold, L. and Kelly, J. (1981), 'Satellite power system: Concept development and evaluation program', *NASA Reference Publication, 1076* .

ESA (2016*a*), 'HEXAFLY international'. [Accessed 20-February-2017].
URL: *http://www.esa.int/Our_Activities/Space_Engineering_Technology/High-Speed_Experimental_Fly_Vehicles_-_INTernational*

ESA (2016*b*), 'HIKARI project homepage'. [Accessed 20-February-2017].
URL: *http://www.hikari-project.eu*

Ewing, G. (1985), *Calculus of variations with applications*, republished 1st edn, Dover, New York.

Ford, J. (2014), 'Airbus tests spaceplane demonstrator'. [Accessed 20-February-2017].
URL: *https://www.theengineer.co.uk/issues/june-2014-online/airbus-tests-spaceplane-demonstrator/*

Fortescue, P., Swinerd, G. and Stark, J. (2005), *Spacecraft Systems Engineering*, 3rd edn, John Wiley and Sons, New Jersey.

Haselbusch, M. and McNelis, N. (1996), Comparison of the continuous freeze slush hydrogen production technique to the freeze/thaw technique, *in* 'The 1996 JANNAF Propulsion Meeting', NASA Lewis Research Center, 201.

Heiliö, M., Lähivaara, T., Laitinen, E., Mantere, T., Merikoski, J., Pohjolainen, S., Raivio, K., Silvennoinen, R., Suutala, A., Tarvainen, T., Tiihonen, T., Tuomela, J., Turunen, E. and Vauhkonen, M. (2016), *Mathematical Modelling*, Springer.

Helenbrook, R., McConarty, W. and Anthony, F. (1971), Evaluation of active cooling systems for a Mach 6 hypersonic transport airframe, part 2, Technical report, NASA.

Hendrickx, B. and Vis, B. (2007), *Energiya-Buran: the Soviet space shuttle*, Springer Science and Business Media, Berlin.

Heppenheimer, T. (2009), *Facing the heat barrier: a history of hypersonics*, Vol. 4232, Government Printing Office, Washington.

Hillje, E. (1967), Entry flight aerodynamics from Apollo mission AS-202, Technical report, NASA.

Hohmann, W. (1994), *Die erreichbarkeit der himmelskörper. Untersuchungen über das raumfahrtproblem.*, by Hohmann, München.

Jenkins, D. and Landis, T. (2003), *Hypersonic: the story of the North American X-15*, Specialty Press, North Branch.

Kirk, D. (1970), *Optimal control theory; an introduction*, Prentice-Hall, New Jersey.

Kirkpatrick, S., Gelatt, C. and Vecchi, M. (1983), 'Optimization by simulated annealing', *Science* **220**(4598), 671–680.

Krotov, V., Gurman, V. and Bukreev, V. (1971), *New variational methods in flight dynamics*, Israel Program for Scientific Translations, Springfield.

Le Goff, T. and Moreau, A. (2013), 'Astrium suborbital spaceplane project: Demand analysis of suborbital space tourism', *Acta Astronautica* **92**(2), 144–149.

Mack, A., Steelant, J., Adirim, H., Lentsch, A., Marini, M. and Pilz, N. (2011), FAST20XX: Achievements on European suborbital space flight, *in* '7th ESA Aerothermodynamics Symposium', 9–12.

Müllemann, A. and Lübken, F. (2005), 'Horizontal winds in the mesosphere at high latitudes', *Advances in Space Research* **35**(11), 1890–1894.

NASA (2012), 'Aero-space plane: Flexible access to space'. [Accessed 20-February-2017].
URL: *https://youtu.be/vBQk6a2aPoQ*

Norris, G. (2016), 'Virgin Galactic's second SpaceShipTwo begins glide flights', *Aviation Week and Space Technology* **178**(25).

Nourani, Y. and Andresen, B. (1998), 'A comparison of simulated annealing cooling strategies', *Journal of Physics A: Mathematical and General* **31**(41), 8373–8385.

Pars, L. (2010), *An introduction to the calculus of variations*, Dover, New York.

Picone, M., Hedin, A. and Drob, D. (2001), 'NRLMSISE-00 model'. [Accessed 20-February-2017].
URL: *https://ccmc.gsfc.nasa.gov/modelweb/atmos/nrlmsise00.html*

Reed, D., Ikawa, H. and Sadunas, J. (1979), An airbreather/rocket-powered, horizontal takeoff tridelta flying wing, single stage to orbit transportation system, *in* 'Conference on Advanced Technology for Future Space Systems, Hampton, Virginia.'.

Reed, R. and Lister, D. (1997), *Wingless flight: the lifting body story*, NASA, Washington.

Rutowski, E. (2012), 'Energy approach to the general aircraft performance problem', *Journal of the Aeronautical Sciences* **21**(3), 187–195.

Salamon, P., Sibani, P. and Frost, R. (2002), *Facts, conjectures, and improvements for simulated annealing*, SIAM, Philadelphia.

Savino, R., Russo, G., D'Oriano, V., Visone, M., Battipede, M. and Gili, P. (2015), 'Performances of a small hypersonic airplane (HyPlane)', *Acta Astronautica* **115**, 338–348.

Schneider, J. and Kirkpatrick, S. (2006), *Stochastic optimization*, Springer, Berlin.

Schweikart, L. and Hallion, R. (1998), *The Hypersonic Revolution. Case Studies in the History of Hypersonic Technology. Volume 3: The Quest for the Orbital Jet: The National Aero-Space Plane Program (1983-1995)*, Air Force History and Museums Program, Bolling.

Shevell, R. (1983), *Fundamentals of Flight*, Vol. 7632, Prentice-Hall, New Jersey.

Springs, C. (1998), 'Ascender spaceplane model flight-tested', *Aviation Week and Space Technology* **149**(4), 29.

Sputnik (2009), 'Tupolev aircraft maker to develop Russia's new strategic bomber'. [Accessed 20-February-2017].
URL: *http://sptnkne.ws/d9ab*

Stevenson, B. (2016), 'Further, faster, more lethal', *Flight International* **190**(5553).

Sänger, E. and Bredt, I. (1944), A rocket drive for long range bombers., Technical report, Translated by Technical Information Branch, BUAER, Navy Department.

Taguchi, H., Harada, K., Kobayashi, H., Hongoh, M., Masaki, D. and Nishida, S. (2014), Mach 4 wind tunnel experiment of hypersonic precooled turbojet engine, *in* '19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference', American Institute of Aeronautics and Astronautics.

Trimble, S. (2016), 'DARPA accelerates interest in hypersonic hybrid', *Flight International* **189**(5545).

Vinh, N. (1981), *Optimal trajectories in atmospheric flight*, Elsevier Scientific, New York.

von Kármán, T. and Edson, L. (1967), *The wind and beyond*, Little, Brown and Company, Boston.

Williams, J. and Vukelich, S. (1981), *The USAF stability and control digital DATCOM*, McDonnell Douglas Corporation, Air Force Flight Dynamics Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Ohio.

Zipfel, P. (2000), *Modeling and simulation of aerospace vehicle dynamics*, American Institute of Aeronautics and Astronautics, Reston.

Zipfel, P. (2011), 'CADAC: multi-use architecture for constructive aerospace simulations', *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* **9**(2), 129–145.